

Programmera Lego Mindstormsrobotar

En introduktion till programmering

Oskar Rosén

28/08-12

oros@kth.se

Introduktion i datateknik (II1310)

Sammanfattning

Denna laboration gav en inblick i programmering då en Lego Mindstorm robot programmerades med hjälp av språket NXC (Not eXactly C) i programmet BricxCC. Med en egen dator och information från föreläsningar samt material från kurshemsidan skulle vi med hjälp av parprogrammering lyckas lösa uppgiften att få roboten att utföra de specifika uppgifter som erhöles.

Då laborationen var en introduktion till programmering var uppgiften väldigt enkel för en som programmerat sedan tidigare och gav därmed inte så mycket för mig.

Vi lyckades genomföra denna laboration och jag presenterar i denna rapport resultaten av en lyckad och genomförd laboration.

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	7
Referenser.....	7
Bilagor	8

1. Inledning

Laborationen gick ut på att introduceras till programmering genom att modifiera redan skriven kod så att roboten utför dess specifika uppgifter som vi skulle programmera den att göra. Genom att använda sig av metoden parprogrammering utvecklas också förståelse för programmering då man har någon att diskutera programmeringen med.

Laborationen introducerar också även studenter till att med ett ingenjörsmetodiskt sätt lära oss dokumentera och analysera.

Vi använde oss av egna datorer som medtogs till laborationen och installerade BricxCC, samt laddade ner drivrutiner till roboten från Lego Mindstorms hemsida. Vi tog del av redan skriven kod, som dock utförde andra uppgifter än de som var givna. Så laborationen gick ut på att ändra koden och om vid behov skriva egen kod för att få roboten att fungera korrekt.

1.1 Bakgrund

Kunskaper om och förståelse för programmering är essentiellt för en student på KTH ICT-skolan. En student måste också förstå och analysera givna funktioner och påståenden, men även besitta kunskapen att skriva egen kod och program. Dessa fundamentala kunskaper måste varje teknolog och ingenjör ha med sig i sitt bagage för att klara av ett framtida yrke där programmering nästan alltid finns med på ett eller annat sätt.

1.2 Syfte och målsättning

Målet med uppgiften var ” 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.”¹

Syftet var att bli introducerad till programmering med hjälp av parprogrammering, då vissa aldrig tidigare har programmerat medan andra har programmerat massor sedan tidigare. Då kan parprogrammering vara till stor nytta då en mindre erfaren programmerare kan lära sig av en mer erfaren genom att kolla på och sedan få hjälp. Genom att felsöka i koden lär man sig att förstå vad det är som faktiskt gör att roboten beter sig på ett visst sätt.

Syftet med laborationsrapporten är att lära sig skriva vetenskapliga rapporter och att använda sig av ett ingenjörsmetodiskt sätt. En utomstående person ska därmed kunna upprepa och återskapa laborationen och dess tillvägagångsätt samt resultat efter att ha läst rapporten.

¹ <https://bilda.kth.se/courseId/8498/content.do?id=19150198>

2. Genomförande

Först gick (jag) igenom innan laborationen allt material som fanns tillgängligt för oss på Daisy så att jag skulle vara förberedd till labben. Jag hade även innan laborationen på min dator installerat BricxCC, IDE för NXC (Not eXactly C, språket vi programmerar robotarna i).

Väl på laborationen installerades drivrutiner från Lego Mindstorms hemsida så att roboten kunde ansluta till datorn.

Vi började med att testa roboten för att se vad det var som den gjorde "fel". Därefter gick jag och min labbpartner igenom koden och letade efter vad som var fel. När vi trodde att vi hade hittat det som var "felet" ändrade vi koden och testade roboten igen. Detta upprepades tills att roboten utförde korrekt de uppgifter som skulle göras.

Uppgiften var:

"Uppgiften består av att installera mjukvara på en PC för kodskrivning och flashning av en LEGO Mindstorms-robot, att testa att köra det program som tillhandahålls vid övningen och sedan att felsöka och modifiera detta program så att roboten på önskat sätt utför de uppgifter som beskrivs vid övningstillfället."²

Dessa uppgifter beskrivs nedan:

- Följa den svarta linjen (svart tejp som lotsad bana på ett bord).
- När den körde in i lådan (slutet av linjen) skulle den stanna
- När den stannat skulle den skriva ut gruppmedlemmarnas namn på skärmen
- Därefter spela en liten trudelutt
- Sedan tillslut skulle den "dansa" lite efter att den spelat upp trudelutten.

Vägen var nästan formad som ett u och i slutet av vägen var en låda som roboten skulle åka in i och därmed stanna.

Parprogrammeringen gick till så att vi båda två turades om att skriva/ändra kod och den person som inte skrev kod satt bredvid och hjälpte till med idéer och förslag på hur problemet skulle lösas, det vill säga att det var inte bara den som skrev kod som gjorde något utan vi arbetade aktivt tillsammans hela tiden.

² <https://bilda.kth.se/courseId/8498/content.do?id=19150198>

3. Resultat

Radnummer	Ny kod	Kommentar
2	#define SpeedSlow 20	Ändrade hastigheten när roboten kör långsamt. Tidigare var hastigheten 80.
3	#define SpeedFast 40	Ändrade hastigheten när roboten kör fort. Tidigare var hastigheten 100
35	string groupMembers[] = { "Oskar", "Filip"};	Våra namn i listan groupMembers
45	TextOut(0, (LCD_LINE2 + (8*i-16)), names[i]);	Plustecken istället för minus, annars skrivs det på föregående rad. Nu skrivs namnen upp på ny rader.
59	Wait(SEC_3);	Vänta 3 sek istället för 20 (var bara jobbigt med 20 sek och vänta)
68	lightIntensity = SensorRaw(IN_3);	Rätt port för ljussensor, var tidigare (IN_1), vilket var fel port.
87	OnFwd(OUT_A, SpeedFast);	Sakta på linjen, köra fort tillbaka mot linjen
94	OnFwd(OUT_A, SpeedSlow);	Sakta på linjen, köra fort tillbaka mot linjen
114	//dance()	Kommenterade bort funktionen dance då den inte ska kallas på förrän i slutet. Innan var det roboten gjorde allra först innan den började leta efter den svarta linjen.

4. Analys

Vi klarade av laborationen ganska fort (ca 30 min) då vi båda hade programmerat sedan tidigare. Därför var denna laboration ingen utmaning för oss, utan vi fick roboten att utföra det den skulle.

Till en början var funktionen dance() kallad på så det första den gjorde när den sattes igång var att börja dansa.

Det var fel port som var angiven för ljussensorn så därför fungerade inte den till en början. Vi valde att köra roboten sakta istället för fort då ljussensorn annars inte hängde med, vilket resulterade i att roboten körde rakt fram istället.

Genom att byta tecken (" $<$ " och " $>$ ") på raderna 82 & 85 resulterade i att roboten strävade efter att åka efter den svarta linje, istället för att svänga bort från linjen.

- dance() – så att roboten "dansar"
- printNamesToScreen() – skriver ut på robotens display arrayen groupMembers[] där våra namn var insatta.
- readTouchSensors() – känner av om den kör in i något, spelar då upp trudelutten och och kallar på på tidigare nämnda funktionen dance().

5. Diskussion

- Några av problemen vi stötte på var att ljussensorn inte fungerade, men detta löstes genom att ange korrekt port för ljussensorn.
- Ett till problem var att bilens hastighet var alldeles för hög vilket resulterade i att ljussensorn inte hann med och bilen fortsatte rakt fram istället, detta löstes genom att bara sänka hastigheten.

Då jag programmerat sedan tidigare kan jag inte påstå att jag direkt har lärt mig något av denna laboration. Men för en person som endast programmerat väldigt lite eller inte alls så skulle nog denna laboration vara givande, men inte för en person som jag som har programmerat sedan tidigare. Dock var laborationen väldigt rolig då vi efter att ha genomfört laborationen fortsatte att utveckla koden så att roboten dansade annorlunda och spelade upp annan musik samtidigt som den dansade.

Dock tycker jag att tillvägagångssättet (parprogrammering) var väldigt bra att använda sig av då man har någon att diskutera med, och alla får prova på de olika rollerna.

Jag tyckte att BricxCC fungerade bra att använda för att skriva kod i. Det gick bra att ansluta roboten när man väl hade installerat drivrutinerna som strulade lite.

Referenser

Här refererar du till de källor du använt dig av när du skrivit rapporten, t.ex. Labb-PM eller internetlänkar.

- <https://bilda.kth.se/node.do?id=19464167>- II1310 Föreläsning 2
- <https://bilda.kth.se/node.do?id=19464168>- II1310 Föreläsning 3
- <https://bilda.kth.se/courseId/8498/content.do?id=19150198> - Labb PM
- <http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html>- NXC API

Bilagor



Oskar Rosén

Hjälp! Sök 

[KTH:s startsida](#) > [Social](#) > [Dagbok](#)

Dagbok

I din dagbok kan du reflektera över dina studier och din personliga utveckling. Du kan använda sökverktyget för att hitta tillbaka till gamla reflektioner.

Läs mer om hur du kan använda dagboken för kontinuerlig karriärutveckling under din studietid.

Skriv en egen anteckning ...

Egen anteckning | 23 augusti kl. 10:49

Har man programmerat sedan innan är det inte svårt, men har man aldrig programmerat innan kan uppgiften vara ganska svår.

Skriv en kommentar...

Skärmdump från KTH Social