

KUNGLIGA TEKNISKA HÖGSKOLAN

Programmering av legorobot i NXC

Lucas Manfredh

2015-08-26

lucasma@kth.se

Introduktionskurs i datateknik II0310

Sammanfattning

I och med all teknik som finns i dagens samhälle blir programmering bara större och större. Då mer avancerade teknik skapas blir koden bara svårare och svårare och det är därför viktigt att kunna hantera och felsöka kod. Vi har därför fått uppgiften att få en redan färdigskrivna kod i språket NXC att fungera felfritt. Koden ska sedan användas för att få en robot att skriva ut våra namn på en skärm och sedan kunna följa en svart linje på golvet. Målet med uppgiften är att ge oss en introduktion till ämnet programmering i form av parprogrammering.

Laborationen började med att vi ladda ner programmet BricxCC samt drivrutiner. Vi fick även en redan skriven kod med ett fåtal fel som vi skulle hitta och rätta till. För att lösa detta använde vi oss av en metod där vi först testade roboten, letade efter fel och försökte sedan hitta hur dessa skulle fixas i koden. Slutligen fick vi roboten att fungera som planerat och den kunde då både skriva ut våra namn och följa en svart linje på golvet.

Problemen bestod främst av småfel så som en int array som innehöll namn och därför behövde bytas till string samt ett fel i y led då våra namn skrevs ut. Sen fanns även problem där koden ändrade hastigheten på fel motor och gjorde att roboten inte kunde följa linjen.

Laborationen var ett roligt och enkelt sätt för mig att jobba med NXC, ett språk jag aldrig tidigare programmerat i.

Innehållsförteckning

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

1. Inledning

Vi har nyligen utfört en laboration med uppgiften att felsöka en redan skriven kod som sedan skulle användas på en legorobot. Detta för att ge oss en kort introduktion till ämnet i form av parprogrammering.

1.1 Bakgrund

I dagens samhälle används programmering till allt möjligt. Det kan handla om allt från Tv apparter och mobiler till kreditkort eller robotar. Oavsett område krävs en stor förståelse av hur programmeringsspråket är uppbyggt och fungerar.

Vid programmeringen av dessa kan koden bli väldigt komplex och risken för ett litet fel som får en stor påverkan på resultatet är stor om man inte är noggrann. Felsökning är därför en viktig del då man programmerar och är nödvändigt för att koden ska fungera.

1.2 Syfte och målsättning

Syftet med uppgiften är att på ett roligt och relativt simpelt sätt introducera programmering i form av parprogrammering. 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.

2. Genomförande

Labben påbörjades med att läsa igenom det Labb-PM vi fått. Därefter fick vi en simpel men tydlig genomgång till hur språket är uppbyggt och fungerar.

Därefter var det nedladdning och installation av programvara som genomfördes. Filerna "BricxCC" och "Drivrutin LEGO NXT" fanns båda tillgängliga för nedladdning bland kursmaterialet. BricxCC är programmet som användes för att skriva kod och flasha LEGO:s Mindstormsrobot. Drivrutiner krävdes också för att väl hitta roboten när den var ikopplad.

Vi använde sedan BricxCC för att öppna och redigera koden. Felen hittade vi genom att först läsa igenom den redan skriva koden och få en ungefärlig uppfattning av vad varje variabel, for loop och funktion användes till. Därefter startade vi roboten, tolkade vad som hände, eller vad som inte hände men borde hända och gick därefter igenom koden rad för rad och korrigerade det som ansågs nödvändigt för att koden skulle fungera som planerat. Denna metod använde vi oss av om och om igen tills slutligen allt fungerade som det skulle samtidigt som vi skrev ner alla korrigeringar. Slutligen kontrollerade vi med läraren att allt såg rätt ut och laborationen var avklarad.

3. Resultat

När laborationen var färdig kunde roboten skriva ut våra namn på skärmen och sedan använda sig av en sensor för att följa en svart linje på golvet. Detta var precis vad som var målet med uppgiften och resultatet blev alltså lyckat. Nedan följer en tabell av de ändringar som gjordes för under laborationen.

Radnummer	Ny Kod	Kommentar
34	String	Byt int till string
46	TextOut(0,(LCD_LINE2 + (8*i-16)), names[i]);	Byt - till +
94	OnFwd(OUT_B,SpeedSlow);	Byt A till B

98	OnFwd(OUT_A,SpeedFast);	Byt B till A
116	OnFwd(OUT_AB,SpeedFast);	Byt SpeedSlow till SpeedFast
2	#define speedSlow40	80 -> 40
3	#define speedFast80	100 -> 80
35	"Marcus Söderman"	Från 1 till namn
36	"Lucas Manfredh"	Från 2 till namn
76		Flyttade ljusensor från 2 till 3
		Flyttade ena touchsensorn från 3 till 4

4. Analys

Då roboten startades var det tänkt att "Gruppmedlemmar:" följt av våra namn skulle skrivas ut på skärmen. Däremot fanns det några fel i koden som gjorde att det inte fungera som planerat. Till att börja med fanns det en array "Int groupMembers[]". Int behövde här bytas ut till String eftersom våra namn består av bokstäver och inte siffror. Våra namn skulle sedan skickas till en funktion "printNamesToScreen" som i sin tur bestod av en for loop som loppade igenom arrayen. Inne i loopen skulle då namnen skrivas ut med hjälp av TextOut(). Här fanns dock ett fel i y-leds positioneringen som gjorde att varje rad hamnade ovanpå varandra. Detta löstes som man kan se i resultat genom att byta ut ett + till -. Genom dessa två ändringar fungerade textutskrivningen som tänkt.

Nästa problem uppstod då roboten med hjälp av en sensor skulle känna av färgen på golvet och på så sätt kunna följa en svart linje. För detta fanns bland annat variablerna "TopThreshold" och "BotThreshold" samt "SpeedSlow" och "SpeedFast". Dessa användes sedan i "readLightSensor()". Inuti denna funktion fanns två if satser. Den första sänkte hastigheten på motor A (höger) om ljusintensiteten var mindre än TopThreshold, d.v.s. om den inte var på den svarta tejen. Om den däremot var över TopThreshold sänktes istället hastigheten på motor B (vänster). Denna if sats följdes upp av en till som ökade hastigheten på motor A om ljusintensiteten var över BotThreshold, d.v.s. om den var på tejen. Annars ökade istället hastigheten på motor B. Som det står i resultatet blev fel motor ändrad på men det bytte vi till rätt och roboten kunde då följa linjen så tänkt.

5. Diskussion

Som tidigare sagt gick uppgiften ut på att felsöka och redigera en redan skriven kod. Detta för att ge färdigheter i felsökning samt att introducera programmering. Då vi först startade roboten fanns en mängd fel. Ett av felen var att utskrivningen av våra namn inte skrevs ut på skärmen som tänkt. Samtidigt kunde roboten inte följa den svarta linje det var meningen att den skulle följa.

Laborationen har gett mig en kort introduktion till programmering inom NXC. Jag har tidigare bara använt mig av Java och PHP så detta var något lite annorlunda. Jag har fått förståelse för hur variabler, loopar och funktioner skrivs ut och har lärt mig en process för att hantera felsökning i kod jag inte själv skrivit.

Oavsett om jag använder mig av samma språk i framtiden eller inte är fortfarande processen för felsökning av kod lika och jag kommer därför säkerligen få ha nytta av vad jag lärt mig.

Den programvara vi använt fungerade utmärkt under labben. Däremot kan jag inte riktigt säga fördelar eller nackdelar med den då vi endast redigerat kort kod. Samma sak gäller för språket NXC. Eftersom jag inte använt mig av någon typ av C språk tidigare detta annorlunda mot hur kod jag vanligtvis skriver ser ut. Dock håller den en liknande struktur så att lära sig den är relativt lätt.

Bilagor

The screenshot shows a web browser displaying a post on the KTH social network. The browser's address bar shows the URL: <https://www.kth.se/social/home/private/post/bra-och-roligt-labb-felsokning-ar-en-av/>. The page header includes the KTH logo and navigation links like 'Lucas', 'Schema', 'Kurser', 'Program', 'Grupper', and 'Tjänster'. The main content of the post is as follows:

KTH / SOCIAL / DAGBOK / BRA OCH ROLIGT LABB. FELSÖKNING ÄR EN AV DE ...

Egen anteckning
| nyss

[Korrigerade inlägg](#) [Mer](#)

Bra och roligt labb. Felsökning är en av de viktigaste delarna då man programmerar och labben var ett bra sätt att träna på det.

Visa tidigare händelser (1) >

Taggades med Labb. | nyss

Kommentera inlägget ...

... eller skriv ett nytt inlägg

On the right side of the post, there are sections for 'Privata sidor' (Private pages) and 'Portfolio' (Portfolio), both with 'Ny sida' (New page) buttons. A 'Taggar' (Tags) section shows 'Labb'. At the bottom right, there is a 'PAPPERSKORG' (Shopping cart) icon.