

Parprogrammerad LEGO-robot

En introduktionslaboration med
programmering i NXC och fördjupning i ICT-
skolans IT-system

Andreas Hahr

2013-08-31

hahr@kth.se

Introduktionskurs i datateknik II1310

Sammanfattning

I laborationen som rapporten avhandlar gavs uppgiften att felsöka och korrigera ett program skrivet i NXC för att sedan flasha en LEGO-robot med programmet och på så vis få roboten att utföra en given uppgift samt att skriva ett inlägg på dagboken i mina sidor på kth.se. Under laborationen lyckas roboten med sin uppgift och dagboksinlägget skrivdes. Programmets funktioner **main()**, **dance()**, **readTouchSensors()** och **followLine()** analyseras för att påvisa laborantens förståelse av programmet och därefter beskrivs de kunskaper som laboranterna förvärvat under uppgiftens gång.

Innehållsförteckning

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

1. Inledning

Laborationen gick ut på att felsöka och ändra i ett program för att få en LEGO-robot att utföra en given uppgift. Laborationen och labrapporten är en del i en introduktionskurs i datateknik med meningen att ge laboranterna bättre kunskap om de IT-system som används inom ICT-skolan och en introduktion i programmering för vilka det kan vara ett nytt kunskapsområde.

1.1 Bakgrund

Som ny studerande på ICT-skolan är det inte säkert att man tidigare varit i kontakt med programmering och/eller KTHs/ICT-skolans olika IT-system. Därför är en introduktion till dessa två områden, om inte nödvändiga så, en hjälp på vägen inför fortsatta studier.

1.2 Syfte och målsättning

Syftet och målsättningen med laborationen, där även själva laborationsrapporten i sig samt inlämningen av den kan sägas ingå, är att ge en introduktion till programmering och KTHs/ICT-skolans IT-miljö. Laborationens konkreta mål var att få en LEGO-robot att utföra en given uppgift; att följa en svart markering på golvet och vid markeringens slut och kollision mot en vägg skriva ut laboranternas namn på robotens skärm.

Syftet med laborationen i sig var att visa hur enklare program och programmeringsfel kan se ut respektive lösas samt att ge en försmak på parprogrammering och programmeringsspråket NXC.

Syftet med laborationsrapporten är att ge en introduktion i hur rapporter bör vara utformade och syftet med hur rapporten skall visas upp för kursledarna är att testa huruvida laboranterna förstått de delar KTH/ICT-skolans IT-miljö som omfattas.

2. Genomförande

Innan laborationens start lästes en tutorial för hur man programmerar en LEGO NXT robot med språket NXC samt Lab-PM och drivrutinerna för LEGO NXT samt programmet BricxCC installerades. Med BricxCC öppnades ett givet program med en rad olika fel i koden för att undersökas av laborationsparet. Olika delar av programmet och felrapporter vid kompilation av programmet diskuterades och laboranterna turades om att göra ändringar i koden till dess att kompilatorn inte längre hittade några fel. Därefter flashades roboten med det redigerade programmet och robotens uppförande undersöktes. Robotens oförmåga att fullfölja den givna uppgiften följdes av mer diskussion och nya ändringar i programmet, nya flashningar av roboten samt nya testkörning till dess att robotens uppförande tillfredsställde uppgiftsbeskrivningen. Ändringarna som gjordes i programmet skrevs kontinuerligt in i en tabell med radnummer samt kommentar. Några kommentarer om laborationen skrevs ned på dagboken på KTH Social.

3. Resultat

LEGO-roboten klarade av att följa den svarta linjen samt skriva ut laboranternas namn på robotens skärm vid linjens slut och kollision mot väggen. Undertecknad blev introducerad till parprogrammering och lärde sig en del om NXC, LEGO-roboten och ICT-skolans IT-system. Dagboksinlägget på mina sidor på kth.se skrevs och en skärmdump på det sparades.

Rad	Ändring	Kommentar
34	int -> string	listan groupMembers[] skall innehålla laboranternas namn och ej heltal, därför ändras klassen från int (integer = heltal) till string (fritext)
35	1 -> "Andreas"	listan groupMembers[] första värde ändas till från 1 till "Andreas" så att Andreas skrivs ut på skärmen
36	2 -> "Farjana"	listan groupMembers[] andra värde ändas till från 2 till "Farjana" så att Farjana skrivs ut på skärmen
44	names -> groupMembers	listan med namn heter groupMembers och ej names
46	(8 x i - 16) -> (i)	för att inte "Gruppmedlemmar", "Andreas" och "Farjana" ska skrivas på samma rad och överlappa varandra ändras algoritmen för radbyte
76	(IN1) -> (IN3)	Ljussensorn var inkopplad i IN3, inte IN1
115	dance(); -> //dance();	funktionen dance gör att roboten snurrar runt och tappar linjen, det försvårar uppgiften och den kommenteras bort
1	SpeedSlow 80 -> SpeedSlow 60	sänker värdet på SpeedSlow för lägre hastighet och en mer lättkontrollerad robot
2	SpeedFast 100 -> SpeedFast 80	sänker värdet på SpeedFast för lägre hastighet och en mer lättkontrollerad robot
94	SpeedSlow -> SpeedFast	exekveras else ska något annat hända än om if exekveras (roboten ska svänga in mot linjen)
98	SpeedFast -> SpeedSlow	exekveras else ska något annat hända än om if exekveras (roboten ska svänga in mot linjen)

4. Analys

Först körs som alltid **main()** och när **main()** körs berättar programmet vilka sensorer som sitter i vilka ingångar och i vilket "mode" de skall verka, därefter kördes funktionen **dance()** som gjorde att roboten körde i en cirkel. Denna togs bort för att göra det enklare att hitta linjen. Efteråt kunde vi ju bara placera roboten direkt på den. Sedan går båda motorerna igång med hastigheten **SpeedSlow** och **readTouchSensors()** samt **followLine()** anropas och loopas. **readTouchSensors()** loopas tills dess beröringssensorn ger utslag och då skrivs namnen ut på skärmen, **finished** får värdet **true** och loopen bryts. **followLine()** körs tills **finished** blir **true** och när ljusintensiteten är mindre än **TopThreshold** fortsätter motor **A** med hastigheten **SpeedSlow**, annars ökar den till hastigheten **SpeedFast**, den svänger alltså mot motor **B** då ljussensorns värde är över intervallet mellan **BotThreshold** och **TopThreshold**. Samma funktion har **followLine()** på motor **B** fast när ljusintensiteten är över respektive under **BotThreshold**, den svänger alltså mot motor **A** när ljussensorns värde ligger under intervallet mellan **BotThreshold** och **TopThreshold**.

5. Diskussion

Tills roboten skulle testköras gick allt bra. Att förstå hur ljussensorn fungerade tog däremot lång tid och jag är inte säker på att jag ännu förstår. Funktionen **followLine()** gjorde som sagt att när ljussensorn värde var över resp. under intervallet mellan **BotThreshold** och **TopThreshold** så svängde roboten åt ena resp. andra hållet. När jag sitter och skriver laborationsrapporten slår det mig att tejpen kanske var trefärgad och det är enda lösningen jag kan komma på för annars begriper jag inte varför golvet skulle vara ljusare på ena sidan tejpen än den andra... Vi löste dock uppgiften eftersom **if-** resp. **elsesatserna** i **followLine()** i ursprungskoden tilldelade motor **A** resp. **B** samma resp. hastigheter vilket uppenbarligen gjorde att roboten svängde åt samma håll hela tiden oavsett ljussensorns momentana värde.

Målet att bekanta sig mer med programmering och skolans IT-system får sägas ha uppnåtts. Vi lyckades med själva uppgiften och eftersom bilda och mina sidor på kth.se använts innan och under laborationen gång samt vid själva rapportskrivningen och rapportinlämningen har vi fått bekanta oss med dessa delar av IT-systemet.

Jag har förutom detta även lärt mig mer om hur LEGO-roboten fungerar och om problem som kan uppstå om kunskaperna kring hårdvaran som program skrivs till är bristfällig. Jag har även fått träning i rapportskrivning!

Kunskaperna jag från kursen har förvärvat kring skolans IT-system kommer jag ha mycket nytta av vid mina framtida studier. IT-systemet känns onödigt krångligt med alla dess olika avdelningar som många har överlappande funktioner och det är kännat skönt att ha fått en bättre överblick över dessa.

De få nya kunskaper jag fått i programmering och om språket NXC tror jag däremot inte kommer att komma till någon större nytta men jag antar detta är sekundärt. Något måste man ju laborera om en rapport ska skrivas och lämnas in!

Referenser

Labb-PM

Bilagor

