

PROGRAMMERING AV LEGO MINDSTORM NXT

EN LABORATIONSRAPPORT OM PROGRAMMERING AV HÅRDVARA I NXC.

Ana Katharina Vasiljevic.

TKOMK, 180p.

anava@kth.se.

II1310 Introduktionskurs i datateknik 1,5 hp.

03.09.2013. Stockholm, Sweden.

Sammanfattning

I denna laboration tilldelades uppgiften att få en LEGO Mindstorm-robot att följa en svart bågformad linje som ledde från punkt A på en vägg till punkt B. När roboten nått punkt B och krockat med sin främre part i väggen skulle följande aktivitet ske; motorerna skulle stängas av, en kort melodi skulle spelas upp och gruppmedlemmarnas namn skulle exponeras på robotens skärm.

Denna rapport behandlar stegvis processen för programmering av detta kommando via programmeringsspråket NXC (Not eXactly C). NXC är speciellt framtaget och anpassat till programmering av Mindstorm robotar. För att underlätta skrivandet av NXC användes BricxCC, ett program som kan beskrivas som ”textredigerare” då det bland annat underlättar redigerande av språket samt hjälper till med att ladda ned- samt överföra det färdiga programmet till roboten. Uppgiften löstes i par om 2 med hjälp av en dator installerad med drivrutiner som nämnts ovan samt instruktioner och material från tidigare föreläsningar.

Syftet med labben var att introducera begreppet programmering, att den utfördes i par beror på den korrekthet som krävs i ett ingenjörsyrke. Det är viktigt att inget kodas fel och risken för detta minskar då något granskas och analyseras av ett fler antal personer. Laborationen lyckades utföras inom ramen av det angivna tidsmaximum (4h).

Innehållsförteckning

1. INLEDNING	3
1.1 BAKGRUND	3
1.2 SYFTE OCH MÅLSÄTTNING	3
2. GENOMFÖRANDE	4
3. RESULTAT.....	5
4. ANALYS.....	5
5. DISKUSSION	6
REFERENSER	7
BILAGOR.....	8

1. Inledning

I detta projekt introducerades en enklare variant av programmering då många deltagare var nybörjare inom området. För att underlätta genomförandet fanns det ett Labb-PM med instruktioner till hjälp samt så användes en redan skriven kod. Denna kod bestod dock av ett flertal kommandon vilket resulterade i att roboten inte följde de angivelser vi skulle uppfylla. Parens uppgift blev alltså att felsöka koden och korrigera de falska angivelserna så att roboten sedan kunde uppfylla de korrekta instruktionerna och målet med uppgiften.

Ett av uppgiftens centrala syften var att påvisa hur ingenjörsmetodik genomförs på ett rätt vis via dokumentation och analysering. Det är alltså den faktiska anledningen till varför en rapport av arbetet sedan skulle lämnas in.

1.1 Bakgrund

En programmeringsfärdighet är något av grunden för vad en ICT-student bör ha lärt sig under sin studietid. Denna labb åsyftade att introducera begreppet programmering till nya elever på skolan. Detta skedde via en mindre komplicerad programmeringsuppgift som innefattade basen och grunderna av programmeringsteknik. Varför just programmering lämpar sig så väl för en framtida ingenjör beror på de kreativa, sökande och problemlösande färdigheter som en person i yrkesrollen faktiskt måste besitta. Programmering generellt handlar om att ge en dator ett kommando att genomföra. Detta kommando kan endast genomföras om datorn begripit vad den faktiskt ska göra. Och det är precis vad ingenjören ska klara av – att ge rätt kommando.

1.2 Syfte och målsättning

Det generella syftet med en laboration är att på ett korrekt och strukturerat vis påvisa vad man undersökt och stegvis beskriva hur detta har genomförts. Inom ingenjörsvetenskapen är ett av nyckelbegreppen korrekthet – detta just då en ingenjör exempelvis kan ha i uppgift att bygga en raket. En raket kan endast flyga om man tillämpa korrekta lagar och strukturer. Ett litet misstag kan få ett helt projekt att falla, detta kan demonstreras via en enkel ekvation; $(x^2 + 3 = 19)$, detta stämmer inte om x inte är 4. Med detta vill man påvisa att varje detalj räknas in och spelar en central roll i det stora, totala sammanhanget. Mycket inom ingenjörsvetenskapen handlar även om att utmana och utveckla sina färdigheter genom övningar som kan likställas med ett programmeringsproblem.

2. Genomförande

För att kunna genomföra laborationen krävdes det en inblick i själva uppgiften, detta gavs via ett Labb-PM som fanns tillgängligt under kursmaterial på BILDAS hemsida (www.bilda.kth.se). Under kursmaterial fanns även de drivrutiner och filer som behövdes, att hämta. Däribland BricxCC som är ett program som används för att skriva och redigera ett språk. I detta fall NXC. En färdigskrivna, men falsk, programkod (line.follower) fanns även tillgänglig på BILDA, denna klistrades in i BricxCC för granskning och redigering. För att slutligen kunna överföra programmet som skrevs i BricxCC och roboten krävdes en nedladdning av drivrutiner för LEGO Mindstorm-roboten. Med samtliga av dessa steg genomförda så kunde programmeringsarbetet påbörjas.

För att hitta felen att ändra i den färdiga koden så var det logisk att börja med att ta reda på vilka kommandon roboten faktiskt utförde med den färdigskrivna koden man tilldelats. Efter att ha analyserat och granskat dessa, kunde felen utpekats och därmed ändras. Detta ledde till att roboten tillslut utförde de korrekta angivelserna; Att med hjälp av en ljusdetektor följa den svarta linjen från punkt A till B. Vid punkt B stötte roboten i väggen och stannade till, spelade upp en kort melodi följt av att gruppmedlemmarnas namn exponerades på skärmen.



Figur 1. LEGO Mindstorm NXT.

3. Resultat

Radnummer	Ny kod	Kommentar
02	#define SpeedSlow 20	Justering av (v) när roboten kör sakta.
03	#define SpeedFast 60	Justering av (v) när roboten kör snabbt.
35	String groupMembers[] = {"Ana", "Mira"};	Namn, gruppmedlemmar.
45	TextOut (8*i)	Tidigare; 16 vilket gör loopen fel.
68	Lightintensity = SensorRaw(iN_3);	Rätt ingång för ljussensor(tidigare IN_1).
87	OnFwd(OUT_A, SpeedFast);	(v) =låg på linjen men hög då den söker sig tillbaka.
94	OnFwd(OUT_A, SpeedSlow);	(v) =låg på linjen men hög då den söker sig tillbaka.
114	//dance()	Dance-funktionen ingick ej i målen för utförande.

Tabell 1; Resultat av ändringar i den givna koden.

4. Analys

Det grundläggande kommandot i koden som styrde om roboten följde den svarta linjen var loopen i "followline" kommandot. I denna loop fanns två stycken så kallade "if-satser" som betecknar "sanningen i påståendet" alltså "om detta är sant". If-satserna kontrollerar alltså på olika vis om roboten faktisk är placerad över den svarta linjen eller inte. För att ta reda på om den är det så anropades även kommandot "redLightSensor" som via robotens ljussensorer scannar in och känner av om linjen finns under den. Om linjen har registrerats av roboten kommer den då följas. Problemet med denna linje var dock att den inte var rak, utan bågformad vilket tvingade roboten att svänga. Detta skedde inte på ett korrekt vis genom grundinställningarna så de behövde redigeras. Roboten styrs av två motorer, A & B. Intervallet mellan dessa justerades för att roboten skulle följa linjen helt och svänga in på rätt ställen. Nästa steg var att göra så att roboten stannade då den nått punkt B, detta styrdes via

loopen i "redTouchSensors" som känner av om robotens främre trycksensorer har aktiverats. I detta fall aktiverades de i samband med att de stötte i väggen i punkt B. När den aktiverats känner "followline" kommandot av detta och ändras till "finish" vilket leder till att roboten stannar efter som processen "followline" avbrutits. Därefter sker nästföljande kommando; en melodi spelas och följs av kommandot "printNamesToScreen"- gruppmedlemmarnas namn skrivs ut på skärmen. Detta är kodens slutliga kommando, efter det tar programmet slut och processen är därmed avslutad.

En av svårigheterna med uppgiften var att ange rätt intervall mellan hastigheten på motor A- och B. Dessa var centrala för att roboten faktiskt skulle följa linjen och svänga in på rätt ställen. En annan faktor som påverkade resultatet var huruvida.

5. Diskussion

Att bli introducerad till ett helt nytt och främmande område är på många vis stimulerande och utvecklande. Trots att laborationen må varit svår att till en början begripa, så resulterade den garanterat i många nya kunskaper. Det man bland annat efter laborationsgenomförandet har lärt sig är att hantera programmeringsspråket NXC och komma i kontakt med några av språkets centrala kommandon nämligen; `OnFwd()`, `OnRev()`, `Off()`, `Wait()`. Sedan även att samtliga program startar med en så kallad *task main*- en huvuduppgift som kommer utföras av roboten.

För att lösa ut och redigera de felaktiga kommandon som fanns i den angivna koden var ett bra alternativ att börja med att låta roboten köra programmet med alla fel. På så vis kunde felen enklare identifieras och därefter åtgärdas.

Allt som utfördes i denna laboration är sådant som studenter på ICT-programmet kommer att ha god användning för i framtiden då man som dataingenjör bland annat ska klara av att programmera. Programmeringen fungerar även som en sorts stimulans för hjärnan och utmanar till att lösa problem genom kunskap och kreativitet. Då programmering, liksom andra språk kräver tid och övning så kan en nybörjare inte förvänta sig att förstå sig på allt på en gång. Det som dock kan ses som en fördel med programmering är att det kan liknas en sammanslagning av de kända ämnena matematik och språk. Dessa har vi alla lärt oss via grundskola och gymnasium, alltså blir helheten och grunden med programmering mycket enklare att sätta sig in i när man har en bas att stå på.

Referenser

Mosquera,I. & Eriksson,R. (2013) *III310 Introduktionskurs i datateknik F1*. KTH; Stockholm. Tillgänglig på: [<https://bilda.kth.se/courseId/10164/content.do?id=21060027>]

Mosquera,I. & Eriksson,R. (2013) *III310 Introduktionskurs i datateknik F2*. KTH; Stockholm. Tillgänglig på: [<https://bilda.kth.se/courseId/10164/content.do?id=21060027>]

Mosquera,I. & Eriksson,R. (2013) *III310 Introduktionskurs i datateknik F3*. KTH; Stockholm. Tillgänglig på: [<https://bilda.kth.se/courseId/10164/content.do?id=21060027>]

Benedettelli,D. (2007) *Programming LEGO NXT Robots using NXC*.
Tillgänglig på: [<https://bilda.kth.se/courseId/10164/content.do?id=21060029>]

Övrigt kursmaterial, såsom drivrutiner finns att hämta under kursmaterial på *KTH BILDA*.

Bilagor

Ana Schema Kurser Program Grupper Tjänster

KTH VETENSKAP OCH KONST

Sök bland kurser, pers

KTH / SOCIAL / DAGBOK

Dagbok

I din dagbok kan du reflektera över dina studier och din personliga utveckling. Du kan använda sökverkyget 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 | tisdag kl. 11:10

Robotlaboration, Tis. 27/8 2013

Jag och min laborationspartner Mira lyckades efter ett flertal försök med uppgiften att få en mindstormrobot att följa en tejplinje. Linjen var formad som en halvcirkel och ledde från en punkt av väggen till en annan punkt. När roboten stötte i väggen med framstycket var den programmerad på så vis att det spelades upp en kort melodi följt av att våra namn exponerades på dess skärm. Med tanke på att det var första gången jag programmerade så var det till en början knepigt att förstå dig på samtliga uttryck, vad som skulle skrivas var osv. Sammanfattningsvis så var labben extremt lärorik, jag har insett att programmering kräver mycket övning, det handlar om att läsa på en hel del. Jag ser verkligen fram emot att bli 100x bättre än vad jag är idag, det känns faktiskt superhäftigt att få chansen att lära sig något heeelt nytt och främmande!

Ana Katharina Vasiljevic

Kommentera inlägget ...

KTH Organisation Tjänster

Figur 2. Printscreen av det dagboksinslägget som skrevs efter genomförandet av laborationen den 27 augusti 2013.