

KUNGLIGA TEKNISKA HÖGSKOLAN

Felsökning av NXC program

LEGO-robot

Albin Byström

2015-09-02

albinby@kth.se

II0310 Introduktionskurs i datateknik

Sammanfattning

Stor del av ingenjörssyrket består av testning, felsökning och verifiering. För att testa på och se hur ingenjörssyrket kan se ut består denna laboration av att felsöka ett färdigt program som styr en LEGO-robot. Programmet är tänkt att få LEGO-roboten att följa en svart tejp. Genom felsökning, testning och verifiering uppfylldes målet för programmet. Laborationen visade hur ett program och ett programmeringsspråk kan fungera. Vikten av noggrannhet inom programmering lyftes även fram.

Innehållsförteckning

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

1. Inledning

Problemlösning och felsökning är en stor del av ingenjörssyrket. Det är därför bra att börja träna på det tidigt. För att simulera en sådan situation består denna laboration av att få en färdig kod för en LEGO-robot att fungera med hjälp av problemlösning och felsökning. Målet är att få LEGO-roboten att utföra sitt uppdrag korrekt, att följa en svarttejp, stanna när den åker in i väg och skriva ut gruppmedlemmarnas namn på LEGO-robotens skärm.

1.1 Bakgrund

En stor del av ingenjörssyrket består av implementering, verifiering och felsökning. Det är en viktig del när man utvecklar system eller lösningar. Det är därför bra att ha en inblick i hur det går till att felsöka och verifiera lösningar. Det underlättar även sitt eget arbete att kunna identifiera fel och sedan rätta dem. Det kan hända att man ska felsöka ett program skrivet av en annan person, då kan det vara bra att ha en förståelse om hur felsökning och programmering fungerar.

1.2 Syfte och målsättning

Syftet med laborationen är att få en introduktion till programmering, programmeringsspråk och till tillvägagångssättet för enklare testning och felsökning. Även få lite erfarenhet inom de system som finns vid ICT-skolan.

Målet med laborationen är att få en LEGO-robot att följa en svart tejplinje, skriva ut gruppmedlemmarnas namn på LEGO-robotens skärm och få den spela en liten trudelutt genom att felsöka och ändra i ett färdigskrivet program som är skrivet i programmeringsspråket NXC (Not eXactly C).

2. Genomförande

Började med att sätta upp miljön för att programmera LEGO-roboten som består av en huvudenhet, en ljussensor, en touchsensor och två hjul(hjul A och B). Eftersom att dator som användes var en MacBook Air med OSX och att mjukvaran för att utföra laborationen inte stödjer OSX användes en VM(virtual machine) kallad Virtual Box med Windows 10 installerat. Det gjorde det möjligt att installera drivrutinerna och Bricx Command Center som används för att ändra koden och överföra programmet till LEGO-roboten. Sedan gavs VM tillgång till datorns USB kontroller så Windows 10 kunde få kontakt med LEGO-roboten. Sedan påbörjades testningen och felsökningen av programmet linefollower.nxc. Felsökningen gick till genom att först testköra programmet på LEGO-roboten för att se om det var möjligt att identifiera några fel för att sedan analysera koden och upptäcka fel i programmet. Rätta felet och testa igen. Felsökningen fortsatt så genom att testa och analysera koden tills linefollower programmet fungerade korrekt och LEGO-roboten uppfyllde målet.

3. Resultat

Resultat var att LEGO-roboten utförde sin uppgift korrekt, alltså att den följde den svarta tejp, skrev ut gruppmedlemmarnas namn på skärmen och spelade en liten trudelutt. Ändringarna kan ses i tabellen nedan.

Ändringar i koden:

Radnummer:	Ändring:	Kommentar:
34 - 36	Ändring av datatypen int till string och siffrorna 1,2 till "Albin", "Sara" Ändring: string groupMembers[] = { "Albin", "Sara"};	Funktionen printNamesToScreen tar en string lista som inparameter inte int lista. Vilket gav ett kompilerings fel.
46	Tog bort "-16" Ändring: TextOut(0, (LCD_LINE2 - (8*i)), names[i]);	Namnen skrevs ut för högt upp på LEGO- robotens skärm
76	Fel sensor Ändring: lightIntensity = SensorRaw(IN_3);	Variabeln lightIntensity fick värde från fel sensor
92	Ändring av hastigheten för hjul A från långsam till snabb Ändring: OnFwd(OUT_A, SpeedFast);	När ljussensor känner av både det ljusa golvet och den mörka tejp. Åker hjul A snabbt.
100	Ändring av hastigheten för hjul B från snabb till långsam Ändring: OnFwd(OUT_B, SpeedSlow);	När ljussensor bara känner av det ljusa golvet. Åker hjul B långsamt.
115	Kommenterar bort funktion dance Ändring: //dance();	Fick roboten att hamna på fel bana vid starten.
2	Ändring av SpeedSlow Ändring: #define SpeedSlow 50	Sänkning av lägsta hastigheten för att få LEGO-roboten att justera/svänga snabbare.

4. Analys

För att få LEGO-roboten att uppfylla målsättningen alltså att få den att följa en bit tejp och när den åker in i en väg så ska gruppmedlemmarnas namn skrivas ut på skärmen och det ska spelas en liten trudelutt. Det som styrde hur LEGO-roboten skulle följa tejplinjen var task followLine som använder sig av ljussensorn för att avgöra hastighet för hjul A och B. Genom att ha olika hastigheter på hjul A och B ger det LEGO-roboten möjligheten att svänga. Med hjälp av funktionen readLightSensor så kan LEGO-robotens ljussensor läsas av. Beroende på underlag

så ger sensor olika värden, ett högt värde är lika med mörkt underlag exempelvis den svarta tejpens och ett lågt värde är ett ljust underlag exempelvis det ljusa golvet. Detta i sin tur kontrollerade hastigheten för hjul A och B. Beroende på vad ljussensor läser av justeras hastigheten för hjul A och B.

För att få roboten att stanna användes task `readTouchSensor`, som lyssnade efter att LEGO-robotens touch sensorer skulle aktiveras, vilket händer när roboten åker in i ett hinder. När den känner av det så stannar den hjul AB, skriver ut gruppmedlemmarnas namn på LEGO-robotens skärm och spelar en liten trudelutt.

Laborationen gick bra fram tills att logiken för hur LEGO-roboten skulle hålla sig på tejpens skulle felsökas. Det var förvirrande hur tankegångarna skulle gå, det blev istället att pröva sig fram med ändringar och testa koden för att analysera LEGO-robotens beteende. Det ledde fram till slutsatsen att det inte endast var fel i rörelse logiken utan även i hur stor skillnaden var mellan högsta och lägsta hastigheten. Skillnaden mellan högsta och lägsta hastigheten påverkade LEGO-robotens svängradie. En sänkning av den lägsta hastigheten gjorde att LEGO-roboten rättades sig bättre efter tejpens tack vare den minskade svängradien.

5. Diskussion

Syftet med laborationen var att få en introduktion av programmering, felsökning, och hur några av ICT-skolans system fungerar. Laborationen blev en introduktion i hur ett program och ett programmeringsspråk kan fungera. Det märktes tydligt hur viktig noggrannhet är inom programmering. Räcker med att en liten del av koden är inkorrekt för att programmet ska sluta fungera, vilket gör noggrannhet otroligt viktigt vid programmering. Laborationen gav även en inblick inom felsökning av kod. Att testa ett program för att identifiera eventuella fel, ändra dem och testa igen tills programmet fungerar korrekt. Vilket också en viktig del inom programmering. Felsöknings metoden som användes var att testa, analysera testet, korrigerar och testa igen. Det vill säga att felsökningen började med att testköra LEGO-roboten och undersöka beteendet. Identifiera fel i LEGO-robotens beteende, om fel hittas analysera koden, ändra den felaktiga koden och testköra igen. Stegen upprepas tills programmet får LEGO-roboten att uppfylla målen. Metoden fungerade bra och det var inga större problem att upptäcka vad som var fel med programmet.

Mjukvaran för att hantera och programmera LEGO-roboten funkade bra, även fast datorn som användes var en Mac var det inga problem att använda en Virtual Machine med Windows installerat på för att köra mjukvaran. Det som var förvirrande var hur koden skulle överföras till LEGO-roboten efter som att "download" knappen inte var speciellt tydlig. Annars var det en smärtfri upplevelse att använda mjukvaran.

Referenser

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

Labb-PM(Datum: 2015-08-30) <https://bilda.kth.se/courseld/12708/content.do?id=23767349>

Bilagor

Inlägg Egen anteckning

Hade laboration idag, målet var att få en lego robot att åka längs en svart tejp bit. Vi skulle få ett färdigt skrivet NXC program att fungera genom att felsöka och ändra i koden. Vi felsökte och ändrade i programmet tills LEGO-roboten följde linjen korrekt.

Sluta följa inlägget

Anmäl missbruk