

Introduktion - Programmering

En laboration för att introducera
programmering

Joel Abrahamsson

2012-8-23

joelabr@kth.se

II1310 - Introduktionskurs i datateknik

Sammanfattning

För en ingenjör är det viktigt med programmeringskunskaper och att vara duktig på att samarbeta med andra. Syftet med laborationen är att visa varför detta är viktigt genom att introducera programmering i form av parprogrammering. Uppgiften var att felsöka och rätta ett färdigskrivet program för en LEGO-robot. Detta genomfördes genom att först analysera och förstå koden, diskutera vad som skulle förändras och sedan ändra programkoden. Förändringarna i koden gjorde att roboten utförde uppgiften på önskat sätt och att laborationen slutfördes med lyckat resultat.

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..... | 3 |
| Referenser..... | 3 |
| Bilagor..... | 3 |

1. Inledning

Rapporten tar upp varför det är viktigt för en ingenjör att ha programmeringskunskaper och att kunna samarbeta med andra, då syftet med rapporten är att introducera programmering och betona vikten av samarbete. Målet är att felsöka och rätta ett färdigt program för en LEGO-robot, så att den utför uppgiften på rätt sätt.

1.1 Bakgrund

Programmeringskunskaper är mycket nyttiga för en ingenjör eftersom det ger möjlighet för ingenjören att lösa komplicerade problem med hjälp av en dator. Det är också nödvändigt att programmeraren vet hur man felsöker och testar sin kod för att kunna arbeta effektivt och nå önskat resultat. Samtidigt är det viktigt att ingenjören kan samarbeta, eftersom fler personer brukar arbeta mer effektivt än en enskild person. De kan tillsammans ta fram lösningar på problem och dela upp arbetet på så vis att projekt slutförs snabbt och effektivt.

1.2 Syfte och målsättning

Syftet med laborationen är att ge en relativt enkel introduktion till programmering i form av parprogrammering, för att samtidigt betona vikten av samarbete.¹

Uppgiftens syfte är också att ge träning i felsökning och testning, för att på så vis ge grund till effektivt programmeringsarbete samt visa hur enkla fel i koden kan orsaka stora problem.¹

Vidare är syftet att introducera arbetsgången vid ingenjörsarbete och ge träning och arbetsvana i ICT-skolans IT-system.¹

Målet med uppgiften är att felsöka och rätta ett färdigt program skrivet i NXC (Not eXactly C) så att en LEGO-robot utför sina uppgifter på önskat sätt.²

2. Genomförande

Innan laborationen lästes Lab-PMet på Bilda³ för att ge en bild av vad laborationen skulle gå ut på och vilka syften och mål som fanns med laborationen. Därefter laddades ett arkiv med programvaran ned från Bilda och installerades genom att extrahera arkivet i en mapp på datorn. I samband med detta laddades även programkoden ned från samma plats.

1 Syfte, Lab-PM, Hämtat 2012-08-29, <https://bilda.kth.se/couseld/8498/content.do?id=19150198>

2 Mål, Lab-PM, Hämtat 2012-08-29, <https://bilda.kth.se/courseld/8498/content.do?id=19150198>

3 Lab-PM, <https://bilda.kth.se/courseld/8498/content.do?id=19150198>

Vid laborationens början kopplades roboten in via USB i datorn, drivrutinerna för roboten installerades, programvaran, BricxCC, startades och programkoden öppnades för granskning. Första steget i granskningen var att förstå hur koden fungerade. När samtliga gruppmedlemmar hade förstått koden diskuterade gruppen vad som skulle ändras först. När ett beslut hade fattats om vad som skulle bli första ändringen började gruppen programmera.

Programmeringen utfördes i form av parprogrammering. Den ene av de två gruppmedlemmarna satte sig vid datorn medan den andre satte sig bredvid den första. Den förstas uppgift var att programmera roboten medan den andre gav direktiv och kommentarer. När ungefär 20 minuter förflutit bytte medlemmarna uppgift så att den förste gav direktiv och den andre programmerade. Detta tillvägagångssätt upprepades genom hela laborationen.

Laborationsuppgiften löstes genom att felsöka koden och rätta till de fel som hittades. För det första skrevs gruppmedlemmarnas namn in i en array. Därefter åtgärdades det första felet som gjorde att gruppmedlemmarnas namn skrevs ut på felaktigt vis på skärmen. Vidare åtgärdades felen i den ordning de kom i programkoden.

Under tiden som programkoden ändrades gjordes testkörningar för att ta reda på hur robotens ljussensor fungerade samt för att se hur väl uppgiften gick framåt. De flesta testkörningarna utfördes genom att placera roboten på ett svart streck och se om den klarade av att följa det. Några testkörningar gjordes även när roboten var inkopplad i datorn. Då användes något svart, i detta fall en datorväska, och något med ljus färg, i detta fall bordet gruppen satt vid. De sistnämnda testkörningarna gjordes för att kontrollera att hjulen roterade åt rätt håll för de olika färgerna. Programkoden ändrades så att roboten svängde åt höger när ljussensorn var över ett svart område och till vänster när området inte var svart.

3. Resultat

| Radnummer | Ny kod | Kommentar |
|-----------|-----------------------------------|--|
| 35 | "Joel", "Anton" | Lade till våra namn i arrayen groupMembers |
| 45 | (LCD_LINE2 - (8 * i)) | Ändra mellanrummet mellan namnen vid utskrift. |
| 68 | lightIntensity = SensorRaw(IN_3); | Läs in värde från rätt sensor (ljussensorn) |
| 86 | OnFwd(OUT_A, SpeedFast); | Se till att roboten svänger till vänster när den hamnar utanför svarta |
| 90 | OnFwd(OUT_B, SpeedSlow); | Se till att roboten svänger till höger när |

| | | |
|--------|--|--|
| | | den är på den svarta linjen |
| 97-102 | <code>/** */</code> | Kommentera bort definitionen av <code>dance()</code> |
| 114 | <code>//dance();</code> | Kommentera bort anropet till <code>dance()</code> |
| 115 | <code>//OnFwd(OUT_AB, SpeedSlow);</code> | Kommentera bort anropet till <code>OnFwd()</code> eftersom det är onödigt. |

4. Analys

Med hjälp av ändringarna som visas i resultatet lyckades gruppen att slutföra uppgiften.

Det var inte mycket kod som behövde ändras för att uppnå önskat resultat. Fyra rader rättades till för att roboten skulle bete sig rätt och ett antal rader kommenterades bort eftersom de var onödiga. De tre främsta ändringarna som gjordes var att ändra hur gruppmedlemmarnas namn skrevs ut på robotens skärm, vilken sensor som skulle läsas av och hur snabbt motorerna skulle köra, beroende på om sensorn var utanför eller på det svarta strecket.

Fortsättningsvis var det inte helt nödvändigt att kommentera bort alla de rader som presenteras i resultatet. Definitionen av funktionen `dance()` (rad 97-102) behövde inte kommenteras bort, eftersom funktionen aldrig skulle bli anropad efter att anropet till `dance()` (rad 114) kommenterats bort. På grund av bortkommenteringen av funktionsdefinitionen blir dock programmets storlek något mindre, även om det inte spelar roll för uppgiften.

Den sista raden som bortkommenterades, rad 115, hade inte heller behövt kommenteras bort, men det gjordes för att raden är onödig; roboten kommer att röra på sig i alla fall.

Sammanfattningsvis gick laborationen bra och resultatet var bra.

5. Diskussion

Syftet med laborationen var bland annat att introducera programmering, felsökning och testning. Detta lyckades laborationen göra genom att programkoden som tillhandahölls var i behov av både felsökning och testning. Eftersom programmeringen utfördes i par fick gruppmedlemmarna tillsammans diskutera med varandra för att komma på hur koden skulle rättas till. Detta främjade samarbete, vilket också var en del av laborationens syfte. Tillsammans kunde gruppmedlemmarna arbeta sig fram till uppgiftens mål; att rätta programkoden så att LEGO-roboten betedde sig på önskat sätt.

Vidare var programvaran som användes under laborationen, BricxCC, lätthanterligt och enkelt att förstå. Programmeringsspråket NXC var mycket likt C och var inte svårt att greppa för en med tidigare programmeringserfarenhet, men troligtvis inte för svårt för en utan tidigare programmeringskunskaper.

Avslutningsvis anses vår grupps tillvägångssätt vara ett bra sätt för att utföra uppgiften. Gruppen valde att analysera koden innan koden ändrades. Att förstå koden var det viktigaste, vilket gjorde att gruppen förstod snabbt vad som skulle ändras för att lösa uppgiften. Överlag anser gruppen att uppgiften var intressant och att den utfördes på ett bra sätt.

Referenser

Labb-PM, Hämtat 2012-08-29, <https://bilda.kth.se/course/8498/content.do?id=19150198>

Bilagor

Egen anteckning | 23 augusti kl. 10:36

Laboration Datateknik 23/8 - 2012

Uppgift

Vi fick en NXT-robot med en ljussensor, två sammankopplade trycksensorer och två motorer, som drev robotens hjul. Uppgiften var att felsöka och modifiera robotens kod så att den kunde följa en svart linje och stoppa när den stötte till väggen. När den väl stött i väggen skulle den visa gruppledammernas namn på skärmen. Uppgiften skulle utföras parvis.

Mitt omdöme

Jag tyckte laborationen var både rolig och intressant. Jag tror att uppgiften är bra för nya programmerare eftersom den ger dem en utmaning samtidigt som den inte är allt för komplicerad. Personligen har jag programmeringsvana och tyckte uppgiften var relativt enkel och gick snabbt att lösa.

Det var en begränsning med uppgiften, nämligen att vi inte fick ta bort/lägga till kod. För nya programmerare (vilka var i fokus i den här uppgiften) var begränsningen bra, eftersom ingen behövde oroa sig över att de skulle vara tvugna att skriva mycket ny kod för att klara av uppgiften. För mig, å andra sidan, var det lite trist, eftersom jag ville gärna ta bort lite kod och göra programmet lite enklare.

Kort och gott gillade jag uppgiften och jag tyckte den var väl genomtänkt. Som sagt tyckte jag att uppgiften var på bra nivå för nya programmerare och fortfarande rolig och intressant för mig med programmeringsvana.