

# Legorobot

---

## Redigering av program i NXC

Johan Andersson

2012-08-28

johand4@kth.se

Introduktionskurs i datateknik II1310

### **Sammanfattning**

Den här rapporten beskriver hur en uppgift löstes för att få en LEGO-robot att utföra en given uppgift med hjälp av redigering av ett befintligt program. Vi programmerade i par och redovisade uppgiften när uppgiften var avklarad.

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

## 1. Inledning

Laborationen genomfördes för att få kunskaper i enklare programmering och för att få en inblick i skolans olika hemsidor och med dem kunna ändra ett befintligt program för att få en LEGO-robot att följa en linje.

### 1.1 Bakgrund

Bakgrunden till laborationen var att introducera problemlösning och programmering för oss studenter på ICT-skolan då detta kommer att bli en del av våran kommande utbildning och även inom det framtida yrkeslivet. Laborationen ingick i en introduktionskurs i datateknik som är en förberedande kurs för studerande vid ICT-SKOLAN i Kista.

### 1.2 Syfte och målsättning

Syftet med laborationen var att vi efter laborationen skulle ha lärt oss att använda skolans olika sidor på internet för att leta information och för att lösa givna uppgifter, samt att ge en introduktion till programmering till de som inte hade programmerat innan.

Målet var att lösa en uppgift som gick ut på att få ett färdigt program skrivet i NXC, ett programmeringsspråk som liknar C, att fungera så att en legorobot kunde följa en linje på marken och avsluta med att spela upp en signal samt visa laborationsdeltagarnas namn på dess display.

## 2. Genomförande

Hur genomfördes laborationen och med vilka metoder? Beskriv olika praktiska moment så som genomgång av Labb-PM, nedladdning och installation av programvara, kodgranskning, hur parprogrammeringen gick till, lösande av uppgiften, testkörning osv.

Vi genomförde laborationen med parprogrammering, vilket innebar att vi jobbade i par där den ena programmerade och den andra testade roboten och sedan byttes uppgifterna några gånger. Metoden var ett effektivt sätt att hjälpa varandra och lära av varandra på.

Vår uppgift var att redigera ett färdigt program skrivet för att få roboten att följa en linje på marken, men som man medvetet hade smugit in fel i. Roboten skulle följa linjen, köra in i en kartong och spela en trudelutt och visa våra namn i en display. Till vår hjälp hade vi en Windowsdator med BrixCC installerat, som är ett program som används för att kompilera kod och flasha roboten via USB. Utöver detta behövde vi installera en drivrutin för att kunna flasha roboten.

Vi hade möjligheten att se koden redan innan så ett par saker hade jag redan innan upptäckt, men vi tog det från grunden, steg för steg. Så fort ett fel uppkom så löste vi det på det sätt vi trodde var det smidigaste.

### 3. Resultat

Radnummer	Ny kod	Kommentar
2	#define SpeedSlow 20	Sänkte den "låga" hastigheten
3	#define SpeedFast 50	Sänkte den "höga" hastigheten
35	"Martin", "Johan"	La till namnen som skulle skrivas ut på displayen.
45	TextOut(0, (LCD_LINE2 - (8*i)), names[i]);	Fick texten att placeras rätt på displayen.
68	lightIntensity = SensorRaw(IN_3);	Bytte till rätt sensor till ljussensorn.
82	if(lightIntensity > TopThreshold)	Bytte till > , alltså över TopThreshold
84	OnFwd(OUT_A, SpeedFast);	Bytte till en annan hastighet.
84	Wait(100);	Lät roboten köra "inåt" kurvan lite längre tid varje gång.
88	if(lightIntensity < BotThreshold)	Ändrade från >.
92	OnFwd(OUT_AB, SpeedSlow);	Båda motorerna går "sakta".
114	//dance();	Kommenterade bort dance så den inte startade direkt, den skulle avsluta.

### 4. Analys

Analysera kort resultatet av laborationen. Varför gick det som det gick? Vad gick bra/dåligt? Vilka funktioner/loopar/variabler/villkor låg bakom robotens beteende?

Vi höll på drygt en timme, vi körde fast ett tag då vi inte upptäckte att fel sensor var angiven i programmet till ljussensorn. Annars lyckades vi lösa de problem som uppstod. Vi hade även svårt med att veta om det var över topTopThreshold som roboten tyckte det var ljust. Det löste vi genom att lägga in kod för att kunna se i displayen om den skrev ljust eller mörkt då den körde över den svarta tejen som var linjen. Jag är medveten om att vi säkert kunde ha löst uppgiften både med färre ändringar och på ett snyggare sätt men i överlag är jag nöjd med vårt arbete.

Det våran robot gjorde var att när den körde in på den svarta tejen svängde den inåt medan när den körde på ljust underlag körde den rakt fram, så den åkte hela tiden innanför tejen i kurvan. Man skulle kanske ha kunnat gjort det snyggare så den hela tiden låg på linjen.

### 5. Diskussion

För att kunna lösa liknade uppgifter effektivare skulle man kunna sakta och metodiskt gå igenom koden redan innan för att lösa alla problem. Det var dock svårt nu på grund av att mina kunskaper

inom programmering är begränsade. NXC är ändå ett relativt enkelt språk att programmera roboten med, med hjälp av API:n tar man sig fram trots att man har begränsade kunskaper sedan innan. Parprogrammering va ett kul sätt att jobba på det hoppas jag det blir mer av i framtiden. Något som jag har lärt mig är att det finns hur många sätt som helst att lösa problem inom programmering på. Det finns så klart effektivare och sämre sätt men alla lyckas.

## Referenser

Labb-PM

<http://briccc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html> – NXC API

<https://bilda.kth.se/node.do?id=19150204> – Guide till NXC

## Bilagor

