

KUNGLIGA TEKNISKA HÖGSKOLAN

Programmering med NXC Lego Mindstorm

INTRODUKTIONSKURS I DATATEKNIK (II1310), LABB

Klawi@Kth.se

Kasper Wikman

2013-08-27



Sammanfattning:

Många har sedan lång tid tillbaka varit lite rädda för att programmering är något som bara handlar om att läsa kod och inget annat. Detta är en intressant uppstart för alla dem som ska starta en datainriktad linje som får en rolig uppgift med mycket självreflektion och samarbete. Genom att felsöka ett kodstycke och sedan använda den på en Lego Mindstorm-robot är något som gör att intresse väcks. Roboten ska utföra ett uppdrag som går ut på att följa en svart linje i en halvcirkel och sedan stötta in i en vägg och därefter spela ett ljusstycke samt printa ut gruppens namn. Denna laboration gick superb, vi lyckades få roboten att utföra det vi ville att den skulle göra genom rättningar och egna korrigeringar av koden. Detta lede till att vi fick en inblick och en smak av de ingenjörers metoder som används i det vardagliga livet. Vi fick även en smak av vad för saker som kan hända om vissa förhållanden inte är uppfyllda.

Innehållsförteckning

<i>Inledning</i>	2
<i>Bakgrund</i>	2
<i>Syfte & målsättning</i>	2
<i>Material</i> :.....	2
<i>Genomförande</i>	3
<i>Resultat</i>	3
<i>Analys</i>	6
<i>Diskussion</i>	6
<i>Referenser</i>	7
<i>Bilagor</i>	7

Inledning:

Denna laboration är menad att lära oss grunderna inom programmering, då vi ska lära oss lite lätt hur man arbetar med kod på ett roligt sätt med en typisk ingenjörsuppgift. Uppgiften i laborationen var att lyckas med att få en robot att åka längst med en svart linje i en halvcirkel in i en vägg, spela ett ljudstycke och skriva gruppens namn på skärmen.

Bakgrund:

Det är en par-baserad övning som handlar om samarbete mellan två parter vilket förekommer inom arbetslivet. Laborationen är på ett teknologiskt plan, vilket innebär att en viss dator- och programmeringskunskap krävs. Det gäller att kunna använda kreativa lösningar, men även se de fel som finns i lösningarna för att få fram det man vill. Allt detta är förekommande mer eller mindre i arbetslivet för en ingenjör.

Syfte & målsättning:

- Att ge oss en rolig introduktionsövning in i programmeringen. Då får vi se lite kod, som vi får rätta och se hur våra ändringar påverkar roboten.
- Även för att introducera oss till den arbetsprocess som ingenjörer har och det IT-system ICT skolan använder t.ex. bilda & KTH social.
- Övningen ger en lätt grund i att felsöka kod och på så sätt öka effektiviteten för vår programmering. Genom detta kan vi se att små fel i en kod kan leda till något helt annat. På så sätt kan detta ge eleven en bättre förståelse för att perfektion är viktigt i sin kod.
- Visa studenten att de kan tillämpa programmering på ett intressant sätt, så de inte tror att de sitter och präntar en massa kod hela dagarna.

Material:

Drivrutin till LEGO NXT för Windows, Bricx Command Center, Programming LEGO NXT Robots using NXC, NXC Programmer's guide (API), Monteringsspecifikationer & laptop med windows.



(liknande)

Genomförande:

1. Laborationen startade med att materialet samlades ihop. Så som Lego-roboten, sladd och dator. Sedan installerades och laddades koden och mjukvaran ner.
2. Efter att installationerna av programmen var klara så checkades roboten igenom för att se så att alla sladdar satt bra. Annars kanske det blir stora fel om t.ex. hjulet var kopplat till en sensor. På så sätt såg man att allt var i ordning.
3. NXC-kompilatorn startades upp och tog emot koden. Genom att testköra koden direkt kan det verifieras om koden har något fel.
4. Efter att koden har undersökts, så korrigerades koden för att få den att utföra den funktion som efterfrågades. Målet är att hitta den kod som ger roboten funktionen att följa efter den svarta linjen och korrigerar misstag på ett visst sätt om den skulle röra sig bort ifrån linjen.
5. Efter att allt var korrigerat, tillämpades den kod som hade rättats för att kunna utföra uppgiften korrekt.

Resultat:

Vi fick roboten att följa den svarta linjen i en halvcirkel, med hjälp av en ljussensor. Sedan åka in i en vägg och trycka in två trycksensorer. Då spelade roboten en serie toner och sedan skrev den gruppens namn på skärmen.

Fungerande kod

```
//Definierar hastigheterna som motorerna går i
#define SpeedSlow 20
#define SpeedFast 30
```

```
//Gränsvärden för ljussensorn (ändra ej dessa)
#define TopThreshold 630
#define BotThreshold 600
```

```
//Definierar raderna på robotens skärm
#define LCD_LINE1 56
#define LCD_LINE2 48
#define LCD_LINE3 40
#define LCD_LINE4 32
#define LCD_LINE5 24
#define LCD_LINE6 16
#define LCD_LINE7 8
#define LCD_LINE8 0
```

```
int lightIntensity;
bool finished = false;
```

```
//En liten trudelutt
Tone done[] = {
  TONE_C4, MS_50,
  TONE_E4, MS_50,
  TONE_G4, MS_50,
  TONE_C5, MS_50,
  TONE_E5, MS_50,
  TONE_G5, MS_50,
  TONE_C6, MS_200
};
```

Ikke fungerande kod

```
//Definierar hastigheterna som motorerna går i
#define SpeedSlow 80
#define SpeedFast 100
```

```
//Gränsvärden för ljussensorn (ändra ej dessa)
#define TopThreshold 630
#define BotThreshold 600
```

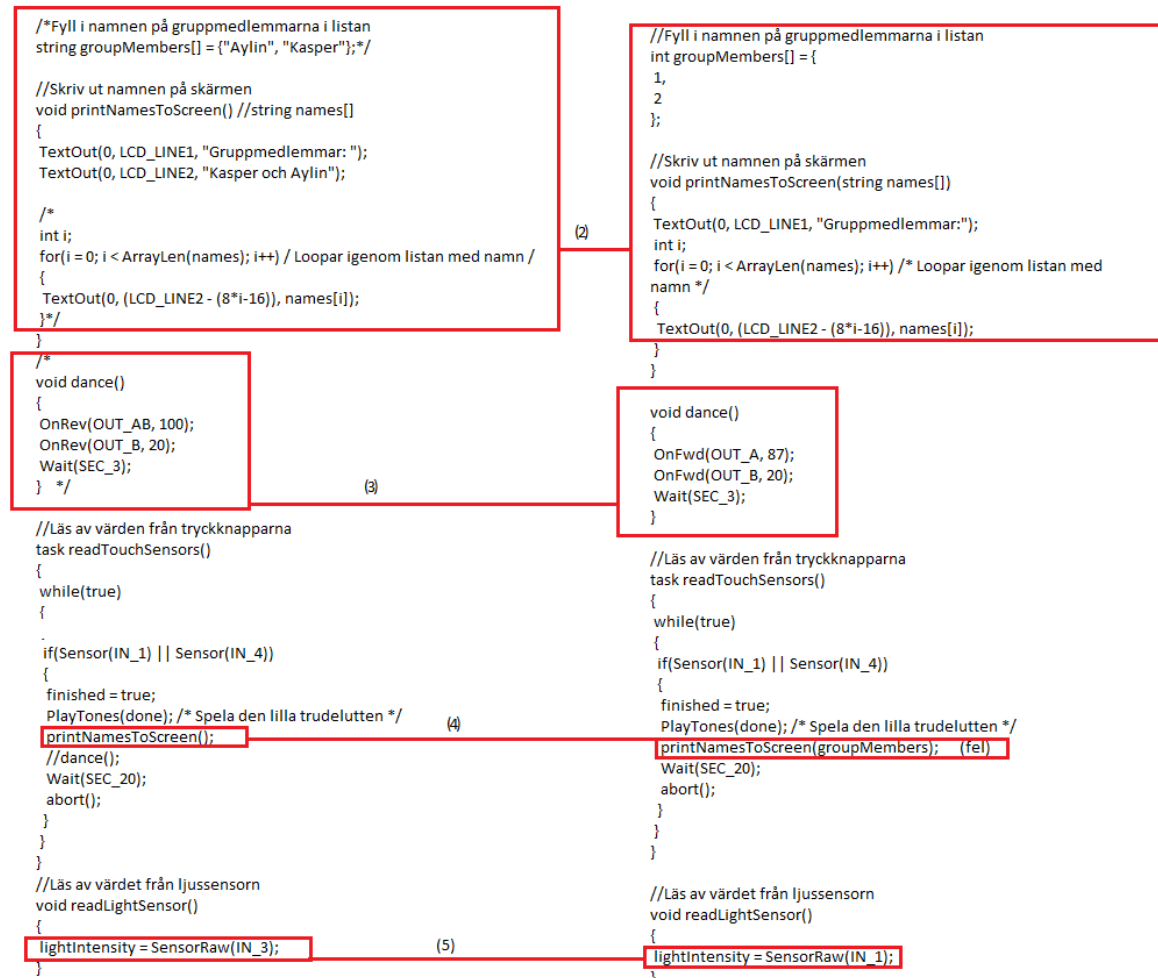
```
//Definierar raderna på robotens skärm
#define LCD_LINE1 56
#define LCD_LINE2 48
#define LCD_LINE3 40
#define LCD_LINE4 32
#define LCD_LINE5 24
#define LCD_LINE6 16
#define LCD_LINE7 8
#define LCD_LINE8 0
```

```
int lightIntensity;
bool finished = false;
```

```
//En liten trudelutt
Tone done[] = {
  TONE_C4, MS_50,
  TONE_E4, MS_50,
  TONE_G4, MS_50,
  TONE_C5, MS_50,
  TONE_E5, MS_50,
  TONE_G5, MS_50,
  TONE_C6, MS_200
};
```

(1)

1. Ändring av robotens hastighet.



2. Int i den andra raden ändrades till string. Vi tog bort stycket i vår kod senare då det inte funkade och ful kodade istället två separata stycken. Vi vet efter senare förklaring att ett ytterligare fel fanns i slutet då det står $(8*i-16)$, minus 16 ska bort för annars printar den inte grupp namnen.

3. Tog bort dance() funktionen då man ville att roboten skulle åka rakt under vissa förhållanden och inte snurra runt i en cirkel konstant.

4. Vi ändrade texten i rutan, egentligen påverkade det inte så mycket. Felet fanns längre upp i koden. Som vi ändrade i början av steg 2 i den andra raden av kod.

5. Ljussensorn var kopplad till IN_3 inte IN_1 därför ändrades detta i koden för att få ljussensorn att läsa av linjen.

```

//Följ linjen!
task followLine()
{
    while(true)
    {
        if(finished == true)
        {
            Off(OUT_AB);
            break;
        }
        readLightSensor();

        if(lightIntensity > BotThreshold)
        {
            OnFwd(OUT_A, SpeedFast);
            OnFwd(OUT_B, SpeedFast);
        }
        else{
            OnFwd(OUT_A, SpeedFast);
            OnFwd(OUT_B, SpeedSlow);
        }

        /*
        if(lightIntensity < TopThreshold)
        {
            OnFwd(OUT_A, SpeedSlow);
        }
        else {
            OnFwd(OUT_A, SpeedFast);
        }
        if(lightIntensity < BotThreshold)
        {
            OnFwd(OUT_B, SpeedSlow);
        } else {
            OnFwd(OUT_B, SpeedFast);
        }
        */
    }
}

//Main-aktiviteten (körs alltid först!)
task main()
{
    Precedes(readTouchSensors, followLine);
    SetSensorType(IN_1, IN_TYPE_SWITCH);
    SetSensorMode(IN_1, IN_MODE_BOOLEAN);
    SetSensorType(IN_3, IN_TYPE_LIGHT_ACTIVE);
    SetSensorMode(IN_3, IN_MODE_RAW);
    SetSensorType(IN_4, IN_TYPE_SWITCH);
    SetSensorMode(IN_4, IN_MODE_BOOLEAN);
    //OnFwd(OUT_AB, SpeedSlow);
}
    
```

(6)

```

//Följ linjen!
task followLine()
{
    while(true)
    {
        if(finished == true)
        {
            Off(OUT_AB);
            break;
        }
        readLightSensor();
        if(lightIntensity < TopThreshold)
        {
            OnFwd(OUT_A, SpeedSlow);
        } else {
            OnFwd(OUT_A, SpeedSlow);
        }
        if(lightIntensity > BotThreshold)
        {
            OnFwd(OUT_B, SpeedFast);
        } else {
            OnFwd(OUT_B, SpeedFast);
        }
    }
}

//Main-aktiviteten (körs alltid först!)
task main()
{
    Precedes(readTouchSensors, followLine);
    SetSensorType(IN_1, IN_TYPE_SWITCH);
    SetSensorMode(IN_1, IN_MODE_BOOLEAN);
    SetSensorType(IN_3, IN_TYPE_LIGHT_ACTIVE);
    SetSensorMode(IN_3, IN_MODE_RAW);
    SetSensorType(IN_4, IN_TYPE_SWITCH);
    SetSensorMode(IN_4, IN_MODE_BOOLEAN);
    dance();
    OnFwd(OUT_AB, SpeedSlow);
}
    
```

(7)

6. Här tog vi bort det originella stycket kod via /*, */ markering. Då utformade vi ett nytt stycke som gjorde de saker vi ville om förhållandet var sant. Vi hade märkt att roboten svängde lite höger så vi kalibrerade den att svänga vänster vid övergång av linjen. Detta gjorde via olika hjulhastigheter.

7. Vi kommenterade bort funktionen att den börjar åka direkt. För vi ville att den enbart skulle köra om den såg den svarta linjen. Här stod även dance(); tidigare men den märkte jag var borttagen i vår slutliga kod.

Analys:

Det som gick bra i labben var att kolla igenom koden och att förstå hur det hela hänger samman. Om instruktionerna hade lästs igenom så skulle man förstå vilka koder som gav roboten en viss funktion.

Genom att observera koden kunde vi se att det fanns lite om detta förhållande stämmer så gör detta, annars använder roboteten denna funktion, den funktionen för ljussensorn fungerade på det visset genom if- och else satser.

Även sådana funktioner som om de inträffar så ska roboten göra detta. Då trycksensorn kom till väggen och spelade upp en ljudsekvens.

Diskussion:

Jag tycker att i denna labb nådde min grupp målen bra. Vi fick prova på en rolig programmeringstillämpning med en logisk och utmanande uppgift. I laborationen fick alla tillämpa saker de kunde till en stor del, vilket var skönt. Jag känner att jag lärt mig lite mer om vad programmering kan användas till och känner mig mer sugen på att starta upp med det hela. Jag tror att detta kan bygga upp ett intresse och öka mina kunskaper. Detta var en ganska grundbaserad labb men jag tror det här ändå på något sätt öppnar upp vyerna för ett allt mer öppet sinne gentemot programmering. Kanske inte tillämpar denna nivå i kommande arbetsliv men tror detta är en grund för vad som komma skall.

Uppgiften var lite problematiskt till en början, då vi trodde allt var fel och var tvungna att gissa lite på att roboten inte riktigt använde vår kod, då vi skulle starta koden i roboten. Då kommenterade vi bort lite saker som var centrala men som ändå körde då vi startade roboten. Det visade sig vara så att roboten inte hade rensats sedan förra gången den användes. Därför körde roboten istället en annan tidigare inlagd kod och det var därför tog det lite extra tid att komma i mål. Efter problemets upptäcktes gick det ganska fort.

Jag tycker programmen som gavs ut var lätthanterliga. Det var lätt att ladda ner och hantera så länge alla instruktioner följdes korrekt. NXC ett helt okej språk, det var inte speciellt svårt. Där gällde det även att läsa i powerpointen men även i texten om roboten så var man väl förberedd.

De ändringar vi gjorde i koden tyckte jag var kreativa och intressanta. Runt en hel del saker kunde kreativiteten bara flöda omkring. Vid slutet av uppgiften kunde roboten spela nästan vilken ton som helst om koden justerades. I min grupp ful kodade vi oss runt ett problem t.ex. och tog och ändrade en annan del helt. Samtidigt får roboten att utföra det den skulle göra. På så sätt blev det en rolig och intressant uppgift.

Referenser:

Lab PM på Bilda, <https://bilda.kth.se/courseId/10164/content.do?id=21060042>
 Programming LEGO NXT Robots using NXC.pdf, på Bilda.

Bilagor:

