

Labbrapport

Programmering av Legorobot NXT

Johan Engberg

28-08-14

jengb@kth.se

II1310 Introduktionskurs i datateknik

Sammanfattning

Att kunna styra hårdvara är en viktig del av en data/el-ingenjörs utbildning. Denna laboration gjordes för att introducera just detta för ovana programmerare på ett lätt och roligt sätt. Detta gjordes genom att felsöka den färdiga koden samt testa de ändringar vi gjort under laborationen. Detta genomfördes under en 3 timmars laboration vid Kungliga Tekniska Högskolan i Kista. Resultatet av laborationen var att roboten utförde sin uppgift och vi fick lära oss hur man interagerar med hårdvara via en dator.

Innehållsförteckning

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

1. Inledning

Rapporten innehåller tillvägagångssätt och analys för felsökning av koden linerfollower.nxc till Roboten LEGO Mindstorm NXT i syfte för att på ett roligt sätt få bättre förståelse för felsökning samt parprogrammering.

1.1 Bakgrund

Bakgrunden till laborationen är för att få en inblick i hur man skriver rapporter och dokumenterar laborationer. Dessa kunskaper kommer behövas under vår studietid på KTH och även efter examen.

1.2 Syfte och målsättning

Syftet med uppgiften är att bekanta sig med programmering och felsökning av kod på ett lätt och roligt sätt. Målet är att felsöka den färdiga koden och få roboten att följa instruktioner och kommandon i språket NXC.

2. Genomförande

Vi började med att ladda ner och installera drivrutiner (Drivrutin LEGO NXT.zip) samt kompilersprogrammet (BricxCC.zip) från bilda.kth.se för att kunna styra LEGO-mindstorm roboten. Vi laddade även ner färdigskrivna koden (linefollower.nxc) som vi laddade upp till roboten. Detta gjorde vi genom att koppla ihop roboten med dator via USB.

Vi började med att granska koden för att få en förståelse om hur roboten agerade och hur programtexten är uppbyggd.

Därefter började vi felsöka koden (linefollower.nxc) och rätta till de fel som fanns.

Efter korrigeringar i koden testkördes roboten för att utvärdera vilka ytterligare justeringar som behövdes göras. Detta genomfördes med hjälp utav parprogrammering, detta innebär att en programmerar medan den andra observerar.

3. Resultat

Efter felsökning och korrigering fick vi det och fungera. Roboten följde den svarta linjen och avslutade med att köra in i väggen som aktiverade sensorerna på robotens armar vilket stoppade programmet och skrev ut våra namn på skärmen samtidigt som den spelade upp en "truddelutt". Detta är ändringarna som vi gjorde i koden:

Radnummer	Ny kod	Kommentar
2	#define SpeedSlow 25	Justering av hastighet för att tidigare hastigheter var för snabb för ljussensorn.
3	#define SpeedFast 45	
34	string groupMembers [] = {	Från int till string, la till namn. Det blev kompileringsfel p.g.a. datatypen.
35	"Johan",	
36	"Calle"	
46	(8*i)	Ändrar ekvationen så att namnen hamnar på rätt radnummer.
77	lightIntensity = SensorRaw (IN_3);	Ändrar input till sensor då den satt på uttag 3.
92	OnFwd (OUT_A, SpeedSlow);	Möjliggör för roboten att svänga vänster och höger.
98	OnFwd (OUT_B, SpeedFast);	
115	// dance();	Bortkommentering av dance();

4. Analys

Funktion för att få roboten att följa den svarta tejpens byggdes igenom att använda en if sats som fick roboten att genom justering av hastighet av motorer på höger resp. vänster hjul få roboten att svänga vänster eller höger beroende av uppmät ljusintensitet framför roboten. När ljusintensiteten var låg pga. att ljuskännaren befanns sig över den svarta linjen ökade hastigheten på höger hjul och sänktes samtidigt på det vänstra viktet resulterade i att roboten svängde till vänster. När roboten svängde av linjen ökade ljusintensiteten vilket ledde till att motorhastigheten justerades så den svängde tillbaka till linjen och detta fick roboten att ta sig fram jämsmed banan. När roboten sedan körde in i väggen aktiverades "task readTouchSensor()" vilket startade funktionen som stoppade motorerna, spelade upp "truddelutten" och skrev ut namnen på skärmen.

Att hitta det felen som från början var i programtexten var inte lätt eftersom programspråket NXC är helt nytt för oss. Med hjälp utan av de labbassistenter som närvarade kunde vi få tips och ledtrådar som tillslut fick oss att hitta de felaktiga avsnitten.

5. Diskussion

Laborationens syfte att få bekanta sig med programmering tycker jag har uppfyllts bra under laborationen. Parprogrammering är ett väldigt bra sätt att lära sig programmera när man är nybörjare. De problem som uppstod beror i mitt fall på för lite kunskap om det aktuella programmeringsspråket men kunde lösas med hjälp av tips och råd av labbassistenterna som närvarade vid laborationen. Programvaran och NXC är utformat på ett sätt så att den ändå är relativt lätt för en person som inte programmerat innan att förstå sig på.

Att få uppleva att se hur skriven text på en dataskärm får roboten att agera väcker ett stort intresse för fortsatt programmering.

Referenser

Materialet som användes för labben hittadas under kursen sida på <https://bilda.kth.se> under flikarna Programmering -> Labb-PM samt Kursmaterial.

Bilagor

Inlägg

Egen anteckning

Programmering av LEGO-Minstorm robot.

Vid dagens laborationstillfälle har jag med hjälp av parprogrammering felsökt programkod till LEGO robot. Laborationen var svår eftersom programspråket är för mig helt nytt men med hjälp utav de labbassistenter som fanns närvarande kunde jag och min kumpan Calle lösa uppgiften!

Johan Engberg skrev inlägget | nyss

Kommentera inlägget ...