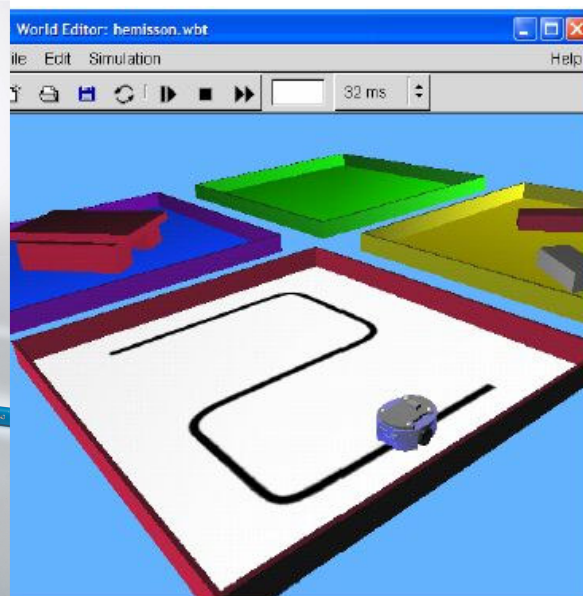


# HEMISSON

## Webots-Botstudio laborationsbok





## Allmänt om Hemisson

Hemisson är en mobil minirobot för utbildning. Liknande enheter förekommer i en mängd sammanhang i dagens samhälle. Det kan vara i form av förarlösa truckar, sk. AGV:er, inom industriella tillämpningar. Det kan vara olika serviceorbotar på tex sjukhus. Det kan vara robotdammsugare, gräsklippare osv. Vi har förmodligen bara sett början på denna utveckling. Alltmer raffinerade robotar tas fram för skräddarsydda ändamål för att effektivisera och underlätta för oss i olika sammanhang...

Hemisson innehåller två motorer som kan driva dess hjul. Genom att aktivera dessa med olika hastighet och riktning kan man få den att flytta sig i den hastighet och kurs man önskar. Den har även ett par lampor och en summer som kan användas för att aktivera ljud och ljus. Detta är de funktioner som finns att aktivera. Dessa aktiveras i programvaran Botstudios händelser i respektive steg. Om man vill kan man skriva villkor för att dessa steg skall utföras. Detta görs då i så kallade övergångar. Villkoren kallas då övergångsvillkor. Villkoret kan vara att en viss tid har förflutit eller att en eller flera givare har ett visst värde. Hemisson har givare nedåt och runtom. Dessa kan få värden mellan 0 och 254. Villkoret kan då vara att en givare skall ha ett värde större än tex 95. En annan gång kan det vara att värdet skall var mindre än 30. På så sätt kan givarna skilja på olika färger och gråskalor mm.

Vi skall lära oss hur vi får detta att fungera. Programmeringsmetoden vi använder i Botstudio heter Grafcet. Detta är en vanlig programmeringsmetod för PLC inom styrtekniken. Programmeringen sker i stort sett i flödesscheman...

---

### SUM TEKNIK AB

Org.Nr.: 556583-6342  
Styrelsens säte: Göteborg  
Bolaget innehar F-skattsedel  
www.sum-teknik.se

### Göteborg

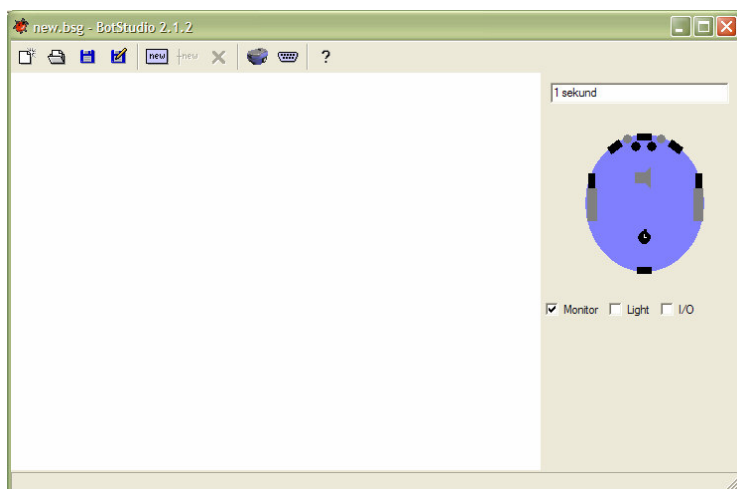
Aschebergsgatan 33  
411 33 GÖTEBORG  
Tel 031-169933  
Fax 031-169933  
Dan@sum-teknik.se

Bankgiro  
5531-6855

# PROJEKT1

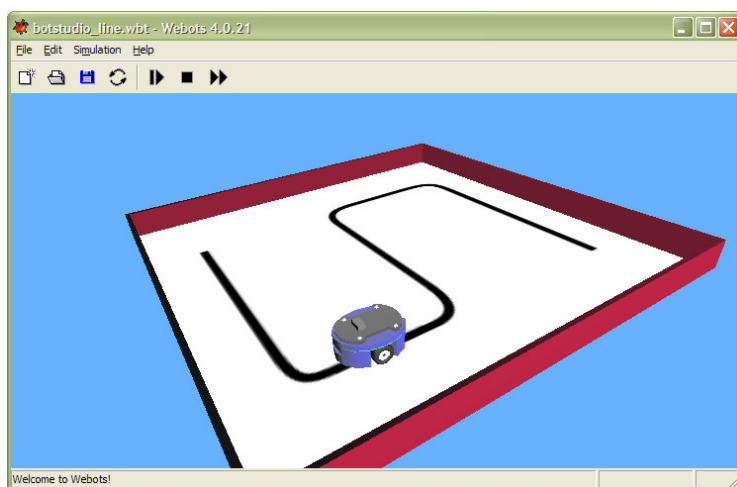
## ”Blinkande lampa”

Gå till den dator där Botstudio finns installerad och starta upp dess programvara (ikon med en liten nyckelpiga på). Ni kommer att få upp två fönster. Det ena fönstret innehåller Botstudio (se nedan).



Det är i detta fönster som själva programmeringen utförs.

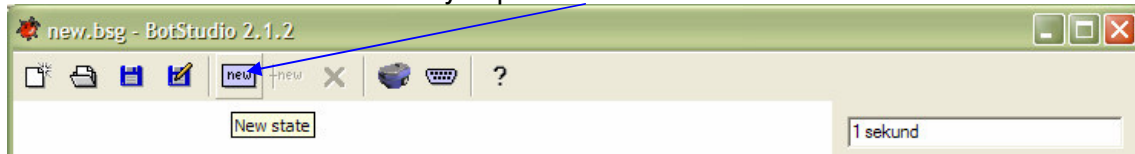
Det andra fönstret heter webots (se nedan) och i detta fönster kan tredimensionella miljöer inhämtas, förändras mm för att passa för simulering av det aktuella programmet.



Då Botstudio startas kommer den att laddas med en förinställd miljö i webots-fönstret (oftast en Hemisson på ett vitt golv med en svart slinga på och med 4 väggar runt om). Botstudio-fönstret brukar vara tomt från början. På så vis kan ni direkt börja programmera era nya projekt. Om ni tycker att det är svårt finns det lösningsexempel att provköra samt digitala videos att titta på för att "få hjälp på traven".

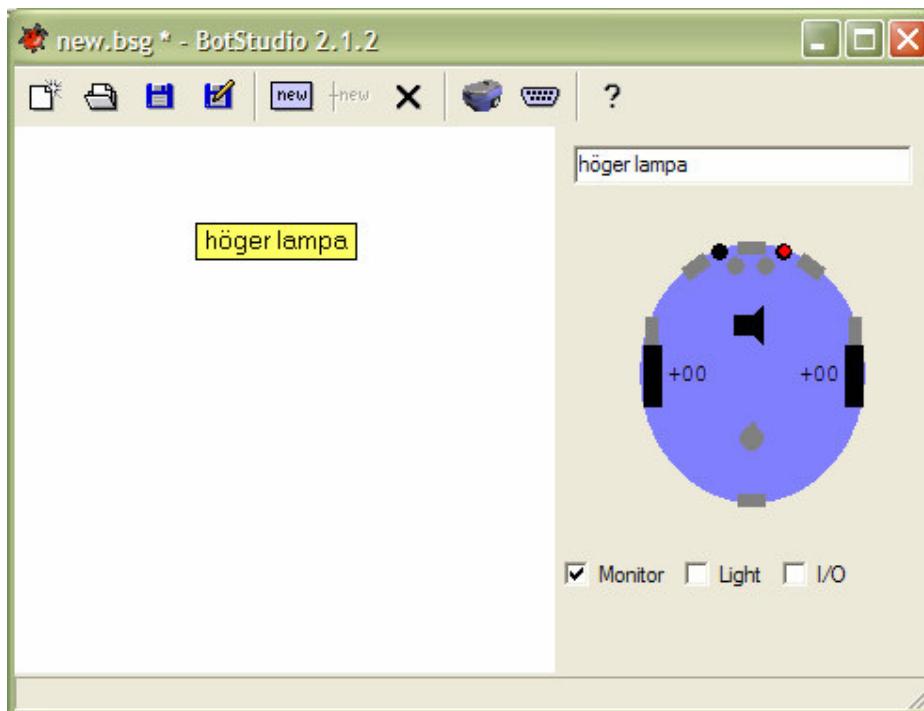
Vi skall nu göra vårt första lilla projekt. Utmaningen denna gång blir att tända höger respektive vänster lampa omväxlande med en sekunds mellanrum.

Aktivera Botstudio-fönstret och tryck på New state-ikonen.



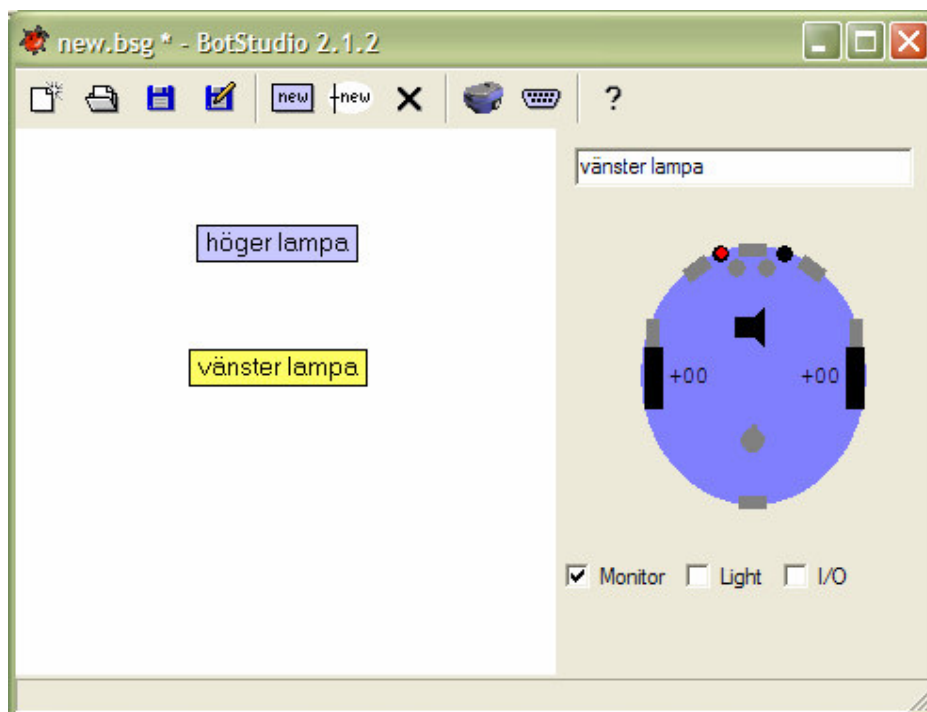
Ett steg hamnar nu i ert program.

För att ange ett lämpligt namn samn vilka händelser som skall utföras i steget används Botstudios högra del.



Skriv in namnet "höger lampa" för steget. Klicka sedan på de högra svarta runda cirkeln för att ange att denna lampa skall tändas. På detta grafiska vis kan man ange vad som skall användas. Vi har alltså en grafisk variant av grafcet, vilket underlättar arbetet väsentligt.

Skapa nu ett steg för vänster lampa på samma sätt.



Om ni vill flytta på ett steg är det bara att markera steget (det blir gult då samtidigt som stegets händelser och namn visas till höger) och dra det med nedhållen vänster mustangent.

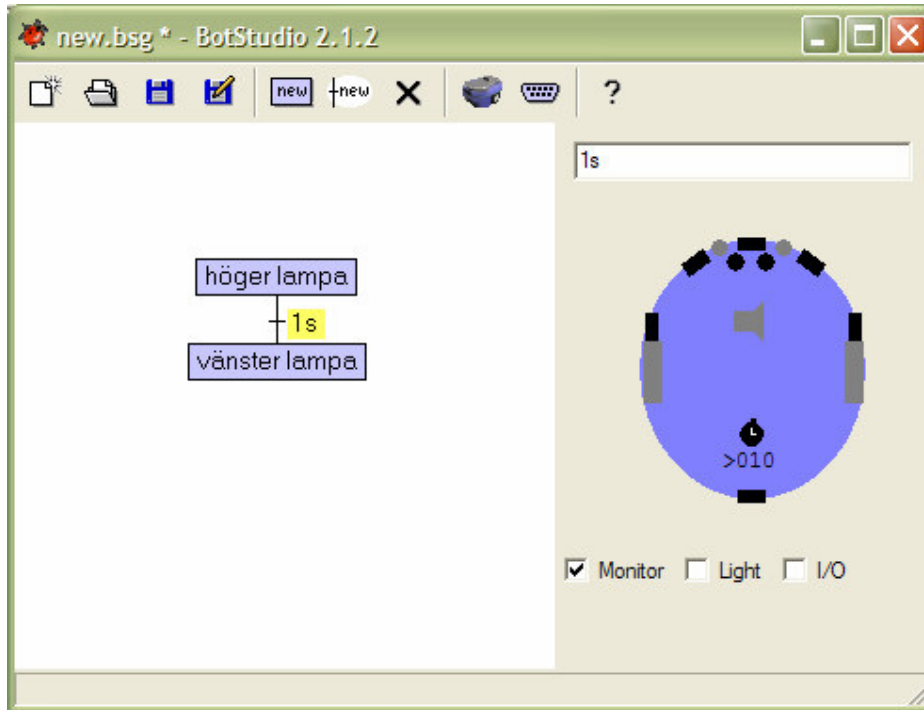
Ordna stegen så att de ligger som ovan.

Nu kan vi skapa ett övergångsvillkor mellan de båda stegen. Detta görs genom att trycka på New transition-ikonen (se nedan).

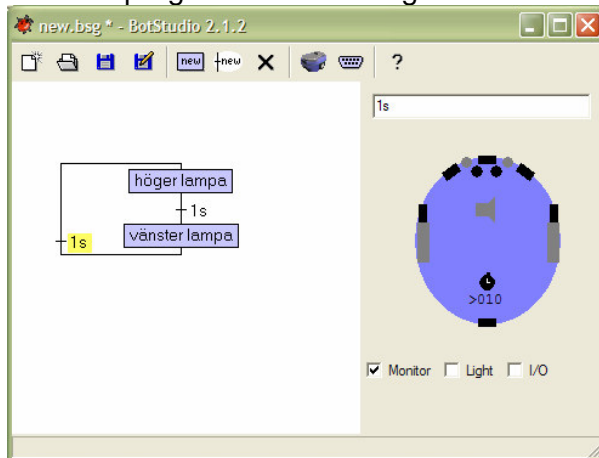


Då denna ikon är intryckt kan man klicka på de steg man vill ha övergången mellan (i vårt fall mellan stegen höger lampa och vänster lampa). Gör detta. Då det är gjort kan man på samma sätt som i stegen ange vad övergången skall heta samt vilka villkor som skall vara uppfyllda i den högra delen.

Döp övergången till 1s och ange att 10 tiondels sekunder skall ha förflutit genom att klicka på klockan. Notera att ni kan ändra tidsangivelsen genom att klicka på siffrorna för entalet, tiotalet samt hundratalet. På så sätt kan ni ange 1 för en tiondels sekund, 10 för en sekund och 500 för 50 sekunder.



Skapa nu ytterligare en övergång. Denna gång genom att först klicka på vänster lampa och sedan höger lampa. Ge denna övergång samma inställningar. Då det är klart bör programmet se ut ungefär som nedan.



Även övergångarna kan flyttas genom att markera dem och dra dem med musen. Spara nu ert program genom att trycka på diskett-ikonen och ge ett lämpligt namn och sökväg.

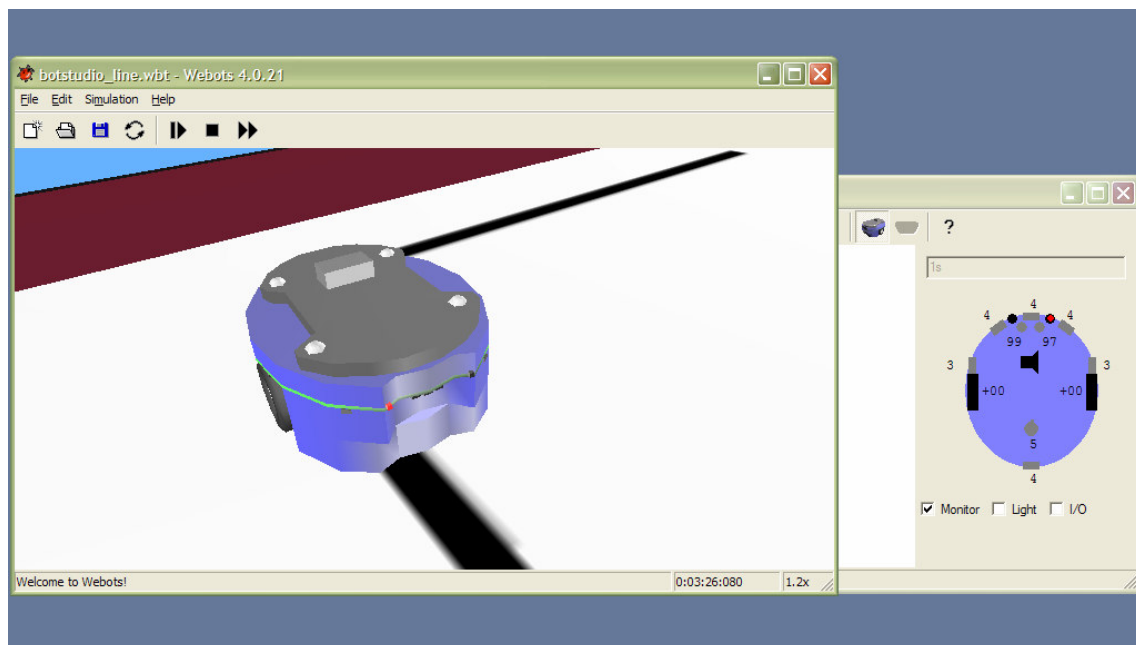
Nu vill vi provköra vårt lilla program. Vi börjar med att provköra genom simulering. Detta görs genom att klicka på Simulate-ikonen (se nedan).



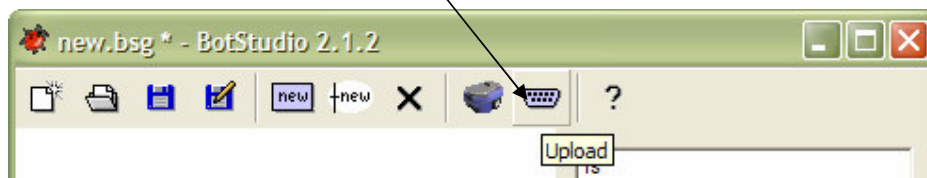
Då detta görs utförs det som programmerats i webots-fönstret. I vårt fall kommer enbart lamporna att blinka på den simulerade Hemisson-roboten. Samtidigt kommer vi att kunna se vilket steg som är aktivt samt lampblinkningar, tider, givarvärden mm i Botstudio-fönstret.

För att kunna se lampblinkningen i Webots-fönstret måste vi lära oss lite vyhantering. Ni kan zomma i webotsfönstret (då et är kativt) genom att använda mushjulet (ni går då stegvis närmare respektive längre ifrån). Ni kan vrida på vyn genom att hålla nere vänster mustangent och föra musen åt olika håll (prova er fram så att ni förstår hur det fungerar), på samma sätt kan ni flytta vyn i sidled (panorera) genom att hålla nere höger mustangent (prova även detta).

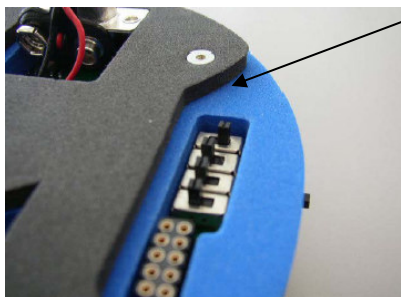
Ni bör kunna få en vy som nedan.



Nu då vi sett att vårt program fungerar i simuleringen vill vi även se att det fungerar i verkligheten. Detta görs med Upload-ikonen (se nedan).



Innan ni gör detta måste ni ansluta den tillhörande nollmodems kabeln till Hemissons 9-poliga D-subkontakt. Ni måste dessutom se till att endast dess översta switch (av de 4 switcharna) är i högerläge (switchen närmast D-subkontakten). Switcharna ställs då roboten inte är påslagen (fungera inte annars).



Kontrollera sedan att Pgm/Exec-switchen på Hemissons vänstra sida står i Exec-läge (skall den alltid göra om man inte skriver om hela operativsystemet, vilket inte är aktuellt då man arbetar med botstudio) och slå sedan på Hemisson med On/Off-switchen på den högra sidan (gröna spänningslampan ovanför switchen tänds då om batteriet sitter i och inte är urladdat). Tänk på att det kan ligga ett gammalt program som kör igång hjulen på roboten. Håll därför upp roboten i luften innan ni slår på (den kan åka iväg annars – den kan åka i golvet och gå sönder)!

Då ni slagit på spänningen trycker ni upload (simuleringen får inte vara igång samtidigt)(notera på fönstrets nedersta rad till vänster att kommunikation sker). Om ni vill kan ni provköra detta program med sladden i. Då kan vi följa hur den arbetar. Normalt sett kan man bara göra detta utan att sladden sitter i (den blir ju "fastbunden" av sladden). Då är det lämpligt att slå av upload-ikonen först och sedan stänga av roboten. Jacka därefter ur kabeln och slå på roboten igen. På så vis kan den åka iväg utan att vara bunden till sladden...

Om ni har den trådlösa modulen på roboten kan ni ligga kvar och se hur roboten arbetar via bluetooth. Då behöver man inte hålla på och slå på och av osv.

Om ni stänger av spänningen till roboten då botstudio fortfarande kommunicerar kan programvaran hänga sig. Det brukar då räcka med att starta om programvaran efter att ha stängt av den (ibland måste aktivitetshanterar tas till med ctrl-alt-del). I värsta fall fås ej korrekt funktion igen förrän ombootning skett.

Om ni fått robotens lampor att blinka är ni nu klara med ert första projekt och är mogna för vårt nästa. Grattis!

# PROJEKT2

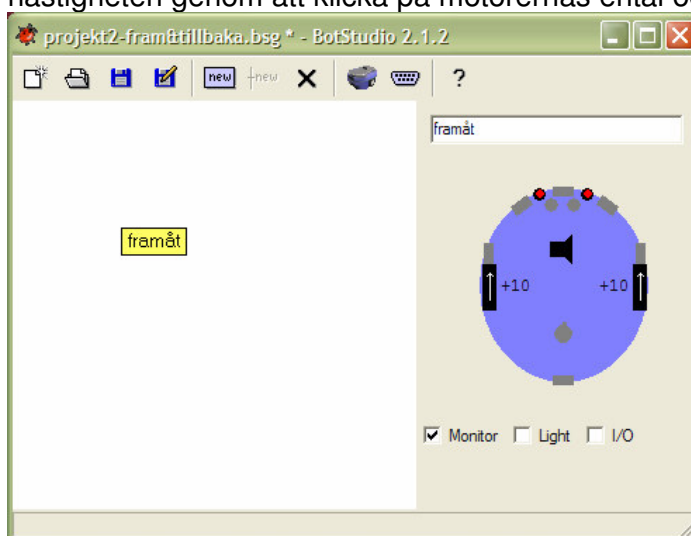
## ”Fram och tillbaka”

Vi skall nu få roboten att börja röra sig.

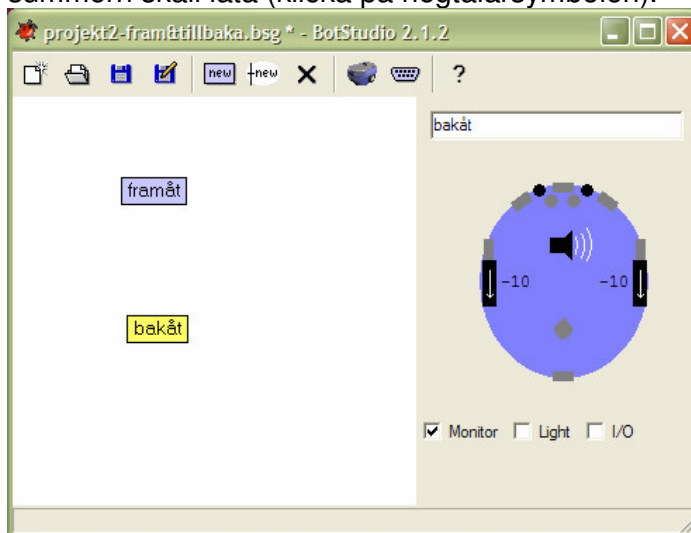
Skapa ett nytt projekt med New graph-ikonen (längst till vänster i botstudio-fönstret).

Skapa ett steg som ni döper till framåt och ange att båda lamporna skall vara tända.

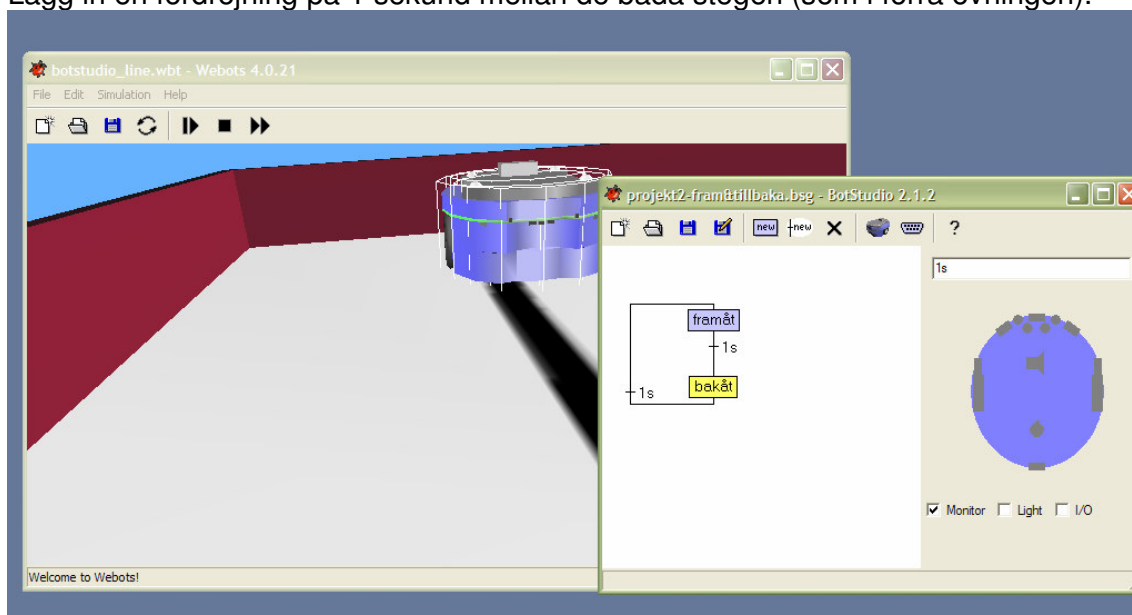
Ange dessutom att båda motorerna som driver robotens hjul skall röra sig framåt (i plus-riktning) med hastigheten 10 (max är 16). Notera att ni kan ändra riktning på hjulen genom att klicka på +/- tecknet. Ni kan på samma sätt som med tiden ändra hastigheten genom att klicka på motorernas ental och tiotal.



Skapa ett steg till som ni döper till bakåt där lamporna ej skall vara till utan istället summern skall låta (klicka på högtalarsymbolen).



Lägg in en fördröjning på 1 sekund mellan de båda stegen (som i förra övningen).



Provkör med simulering och därefter med verklig robot. Fungerade det? Du har programmerat rörelsen på en mobil robot!!

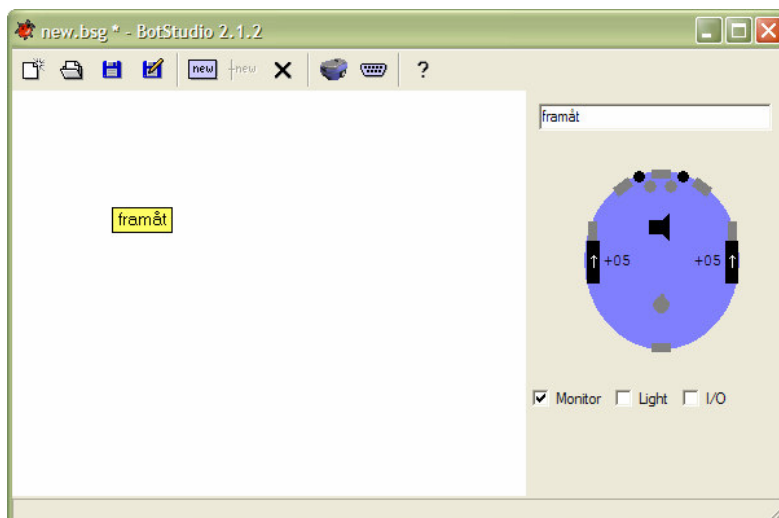
# PROJEKT3

## ”Slingstyrning”

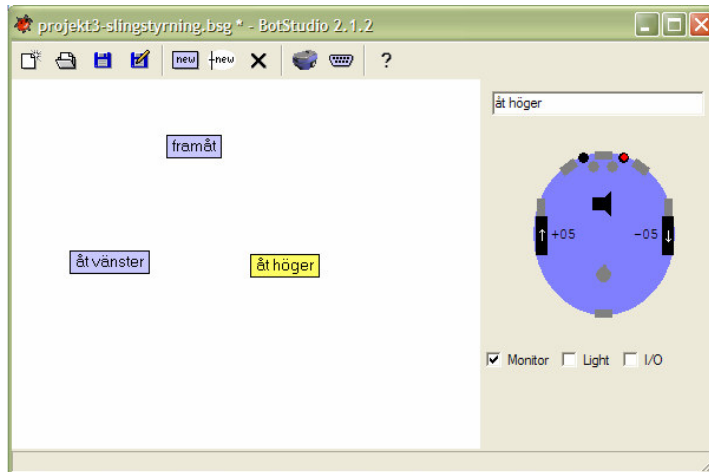
Vi skall nu få roboten att följa en slinga. Detta är en vanlig lösning för styrning av förarlösa truckar (AGV:er – AutomaticallyGuidedVehicle). Dessa kan visserligen styras även på andra sätt – tex med infrarött ljud, kamera eller ultraljud – Det kan även Hemisson, men detta är senare övningar...

I denna övning måste vi börja använda givarna på roboten. I detta fall är det de som är riktade nedåt som behövs. Skulle vi dessutom behöva undvika hinder mm skulle även de på sidan behövas, men detta väntar vi med.

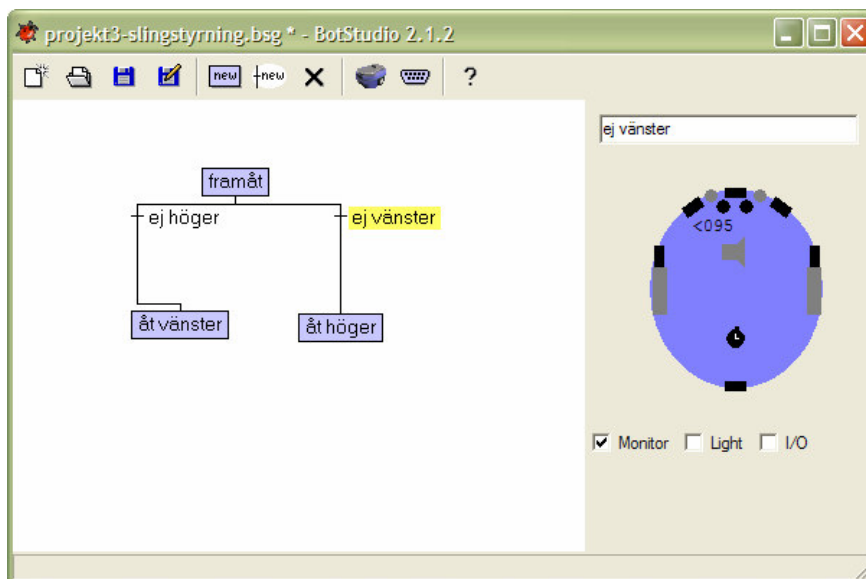
Skapa ett projekt med New Graph-ikonen och lägg in ett första steg där roboten körs framåt med hastigheten +05.



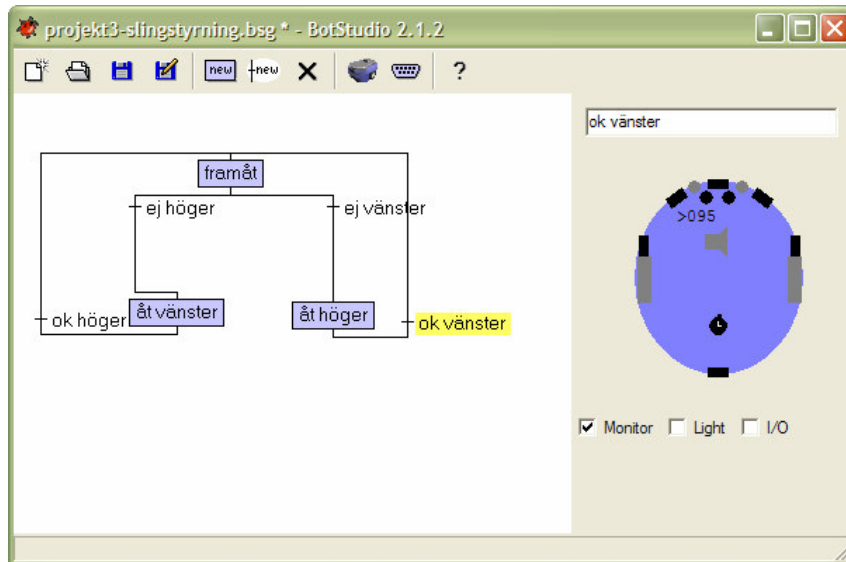
Lägg in två steg där roboten får svänga åt vänster respektive höger (+05 på ena hjulet och -05 på andra hjulet).



Lägg ni i övergångar där roboten får reagera på en lämplig gråskala för slingan (beror på slingans och omgivningens gråskalor). Prova mindre än 95 som tröskelvärde för om "slingan saknas".



Lägg in övergångar för att roboten skall kunna börja åka rakt fram igen då slingan är återfunnen (prova större än 95 som tröskel för detta).



Provkör i simuleringen (världen Botstudio\_line bör vara aktiv när detta provkörs). Om ni behöver flytta roboten (eller något annat), kan ni hålla inne shift och ta den med vänster mustangent. Den flyttas då i planet (utan att flyttas i höjddled). Vill ni vrida på den kan ni ta shift+höger mus på samma sätt. Prova gärna detta.

Ni märker nog att det inte riktigt fungerar som det borde. Tröskelvärdet på 95 är lagom i mitten, men vi har förväxlat värdena för vitt och svart (notera vilka värden som visas vid monitoreringen). Gå därför in och ändra övergången "ok vänster" till mindre än 95 och övergången "ej vänster" till större än 95. Gör samma sak med övergångarna för höger sida.

Provkör simulerat.

Om ni har en slingbana för den verkliga roboten kan ni även prova den.

Ni kan skapa en bana bara med våra pdf-filer för detta (en pdf för raksträcka, en för kurva och en för korsning). Skriv ut flera sådana och sätt samman dem till lämpliga banor (plasta in dem om möjligt). Det kan vara en fördel att ha roboten i en "rink" med väggar runt så att den inte åker i marken (användbart vid andra övningar också).

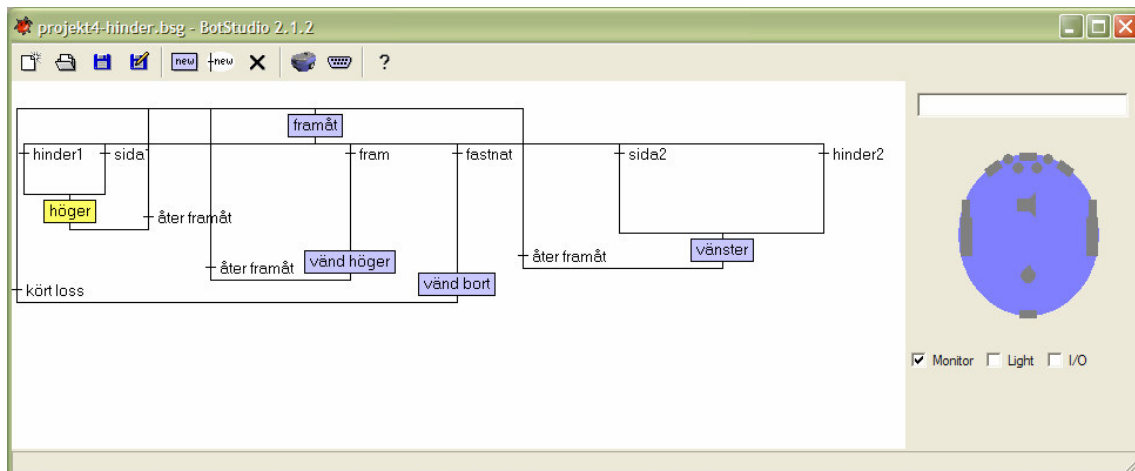
Gör gärna fler projekt för slingstyrning. Vad behöver ni tänka på för att kunna välja väg vid korsningar? Hur kan ni välja väg för roboten om ni använder slingor med olika färg/gråskala?

# PROJEKT4

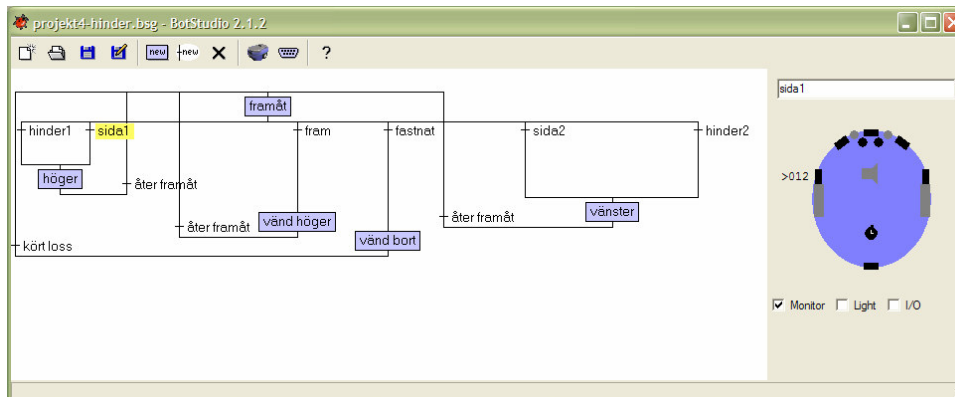
## ”Hinder”

Vi skall nu få roboten att styra undan från hinder. Vi skall i denna övning koncentrera oss på hinder av en och samma färg. I verkligheten kan vi behöva lära den att ta hänsyn till hinder av flera färger...

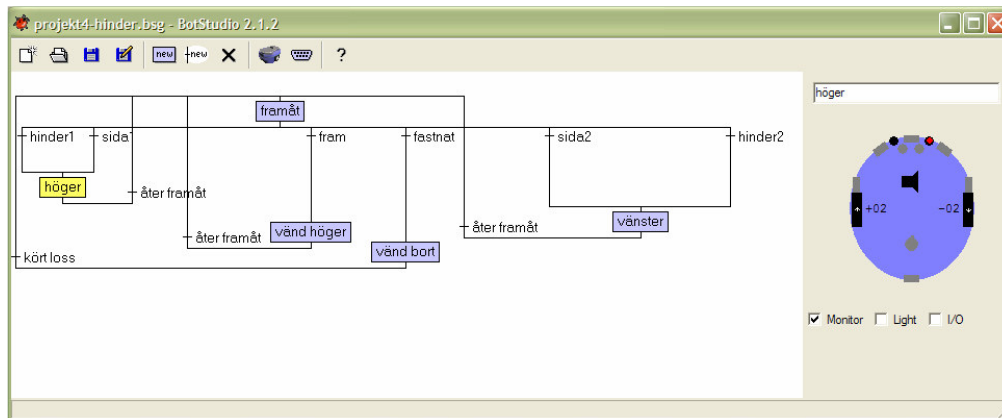
Nedan framgår ett exempel på hur programmeringen för hinder kan gå till. Det hela börjar med steget framåt (som vanligt). Om inget hinder kommer ivägen inom 21 sekunder (anges i övergången ”fastnat”) vänder roboten med steget ”vänd bort” tills den med övergången ”kört loss” förväntas ha kört loss (efter drygt 2 sekunder).



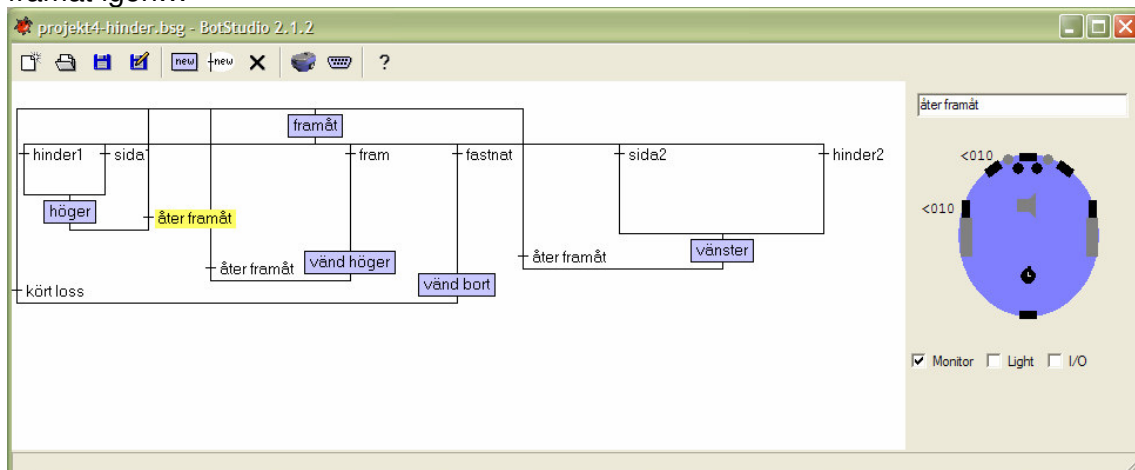
Alternativet är att någon av givarna runt om känner av ett hinder i det angivna gråskaleintervallet (vi väljer värden över 12)(se exempel på sida1 nedan):



Då behöver vi vända oss bort från hindret med lägre hastighet tills de tänkbara givarna (angivna i övergångarna "sida1" samt "hinder1") inte längre reagerar (se nedan).



Då värdena är under de aktuella värdena för hinder kan vi alltså börja aktivera steget framåt igen...



För att få en lämplig simuleringsmiljö laddar ni världen botstudio\_obstacle.wbt till Webots-fönstret. Tänk på att ni kan flytta både roboten och hindren med shift+vänster mustangent och att ni kan vrida dem med shift+höger mus.

När ni provar den verkliga roboten bör detta göras med en rink i vilken ni placerar ut lämplia objekt med lämplig färg. Vill ni göra mer avancerade övningar kan ni lägga in hinder i annan färg. Ni kan dessutom komplettera med slingstyrning där hinder kommer ivägen.

Lycka till!