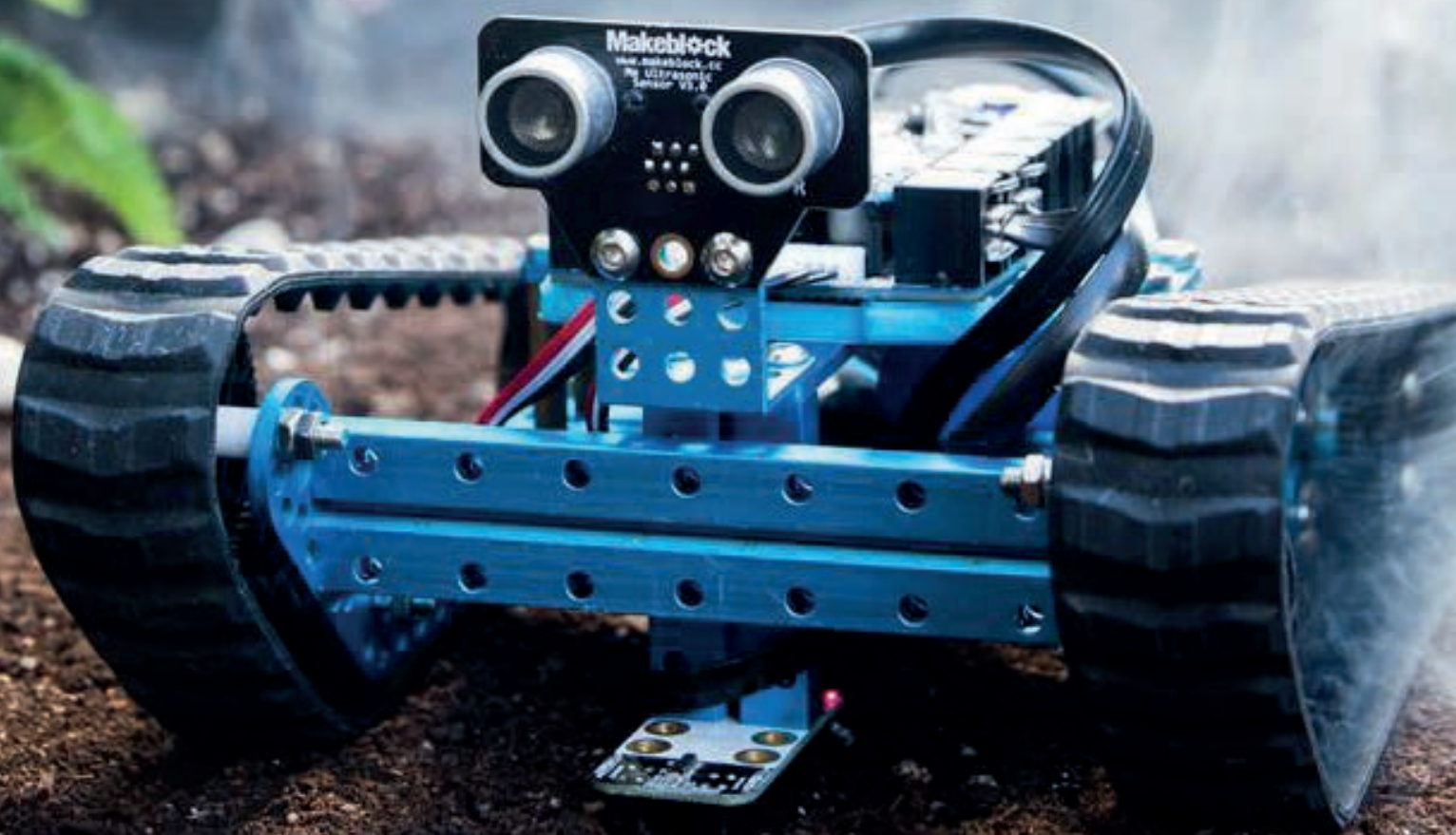


ROBOTPROGRAMMERING

LÄRARHANDLEDNING ÅRSKURS 8



NORRKÖPINGS
VISUALISERINGSCENTER

Till läraren

Välkommen till årets avtalsbesök här på Visualiseringscenter C. Hos oss kommer du och din klass att få chansen att lära er mer om robotar; hur de fungerar, hur de kan programmeras och hur de kan användas.

De senaste årens teknikutveckling har gjort att programmering och automatisering har blivit en stor del av vår vardag. Allt från tvättmaskiner och dammsugare till bilar och huslarm bygger på någon form av digitala styrsystem som vi, utan några jättestora ansträngningar har lärt oss att använda. Idag finns stora möjligheter att automatisera mer eller mindre hela hem. Med detta ökar kraven på datorns prestanda, smidiga användargränssnitt och smarta tekniska lösningar. I många av dagens applikationer finns olika typ av sensorer för att roboten – eller vad det nu skulle kunna vara – ska kunna ta in och processa information utifrån. Det kan handla om röstidentifiering, tryckknappar, röksensorer, bildanalyser, etc. En del av dessa sensorer har så småningom utvecklats till att bli det gränssnitt som vi styr vardagstekniken med. Därför är det viktigt att förstå hur dessa fungerar samt hur ett styrsystem kan ta till sig informationen. För att förstå vår vardag samt att kunna vara med och utveckla den, är det viktigt att utveckla en grundläggande förståelse för hur våra produkter fungerar och hur de produceras. Detta är en av anledningarna att programmering har fått en större betydelse i dagens och i framtidens skola. Något som vi på Visualiseringscenter C har tagit fasta på.

Det besök du kommer att genomföra utgår från två huvudfrågor

- Vad är programmering?
- Hur kan en robot programmeras för att ta hänsyn till sin omgivning?

Frågeställningarna utgår från Skolverkets arbete med programmering i skolan, men också från den kunskap vi själva och våra forskare har av programmering, robotik och artificiell intelligens. På så vis har vi kunnat utforma ett program som både är grundat i dagens skola och framtidens behov; grundat på en stabil grund av praktisk erfarenhet.

Handledningen

Denna text ska fungera som ett stöd i din undervisning. Den innehåller en kort beskrivning av programmering i stort som kan användas som ett förberedelsearbete, men som mycket väl kan användas och utvecklas efter besöket hos oss. Texten innehåller även lite information om den robot ni ska använda samt tips på var ni kan hitta möjligheter att ytterligare förbättra era programmeringskunskaper. Texten är utformad på ett sådant sätt att den kan användas i mindre delar eller som en större helhet.

Sammantaget hoppas vi att detta ska bli en bra grund för ett riktigt bra besök hos oss på Visualiseringscenter C.

Äventyret kan börja!

Besökets upplägg

Ert besök hos oss börjar med att vi tillsammans tittar på en interaktiv föreställning om hjärnan. I föreställningen tar vi upp en del om hjärnans fysiologi och funktion, men också hur våra sinnen tar upp information från världen runt oss. Detta gör vi för att ha en utgångspunkt i hur robotens sensorer fungerar. Efter det kommer vi att göra en gemensam genomgång för att demonstrera hur vår robot fungerar.

Efter föreställning kommer vi att arbeta med robotarna i workshopform. Eleverna kommer att arbeta i grupp, och det kan vara en god idé att i förväg dela in klassen i 10 grupper. Exakt hur ni delar upp grupperna bestämmer ni på egen hand och vill ni dela på tjejer och killar så är det helt okej för oss. Ni kommer att genomföra övningarna med handledning av en av våra pedagoger, men vi önskar att medföljande pedagoger tar en aktiv del av vårt program för att på så vis kunna följa upp besöket i klassrummet.

Vi hoppas att det kan bli en lärorik stund i våra lite dolda delar i huset.

Lite praktisk information

- När ni anländer till kommer ni att tas emot i vår reception. Efter det önskar vi att ni hänger av er ytterkläder en trappa ned. Där nere finns skåp där ni kan låsa in eventuella värdesaker. Tänk på att besöket genomförs tillsammans med minst en annan klass, så försök att snåla lite med utrymmet.
- Då vi önskar kunna hinna med programmet utan onödig stress så är det viktigt att komma i god tid innan besöket. Våra utställningar är tillgängliga direkt på morgonen, så det finns alltid något att göra. Om ni ändå skulle komma lite sent så önskar vi att ni kontaktar oss så att vi vet att ni är på väg (011 – 15 63 00).
- Räkna med att besöket tar ca två timmar, men om ni av någon anledning behöver åka härifrån tidigare så kan vi givetvis komprimera besöket en aning.
- Glöm inte att vi kan erbjuda er skollunch direkt efter besöket. Lunchen kostar 45 kronor och serveras i vår restaurang. Hör av er till vår bokning om ni har frågor angående detta (bokning@visualiseringscenter.se alt. 011-15 63 30).

Roboten – En lång historia

Det är lätt att tro att idén om människoliknande, intelligenta robotar är något som tagits fram under de senaste åren, men det är långt ifrån sant. Redan för 4000 år sedan – i antikens Grekland – skrevs det legender om hur gudarna tillverkade mekaniska tjänare av guld, eller bord som kunde förflytta sig på egen hand. Liknande idéer och berättelser finns att hitta i andra kulturer och de blir allt vanligare vartefter tiden går.

Redan för 2000 år sedan fanns det fungerande mekaniska robotar och gott om skisser på hur de skulle kunna drivas framåt med exempelvis ångkraft. Ingen av dessa robotar kunde utföra speciellt många uppgifter, men redan på 1700-talet finns det exempel på robotar som kan spela musik eller för den delen imitera en ankas ätbeteende.

På 1800-talet började man göra försök med att fjärrstyra robotar med hjälp av exempelvis hydraulik eller elektriska system. På 1870-talet kunde man konstruera en fjärrstyrd torped, och ca 10 år senare kunde den första konstruktionen med trådlös styrning börja säljas till det amerikanska försvaret och hundra år senare kunde man landa den första roboten på en annan himlakropp. Roboten har alltså en mycket längre historia än vad man först skulle kunna tro, men fortfarande är grundtanken den samma: att med hjälp av tekniska lösningar underlätta vår vardag.

Vad är programmering?

Precis som vi människor använder språk för att kommunicera så har vi utarbetat olika typer av ”påhittade” språk, symboler och gränssnitt för att kunna kommunicera med våra datorer och robotar. Idag finns det stora möjligheter att kunna prata ganska så direkt med en robot. Vissa robotar kan känna igen tal eller gester, medan andra robotar har enkla knappar som man kan använda sig av. Vissa robotar kan styras med hjälp av symboler på en dator, medan andra kräver lite mer programmeringskunskaper.

Men för att en robot ska fungera så måste någon ha arbetat fram ett grundläggande system för att datorn ska kunna förstå vad vi vill säga till den. Utan ett sådant system kommer datorn inte veta hur den ska bete sig och inget kommer att fungera som det ska. Hur smart en robot än kan verka vara på utsidan så är det alltid en eller flera personer som har utvecklat grunden på insidan. För att veta vilken typ av programmering man pratar om används ibland orden lågnivå och högnivåprogrammering. På de lägre nivåerna hittar man den programmering som ger förutsättningarna för att en robot ska kunna utföra det vi vill att den ska göra, medan de högre nivåerna ger oss möjligheter att kommunicera med den och ge den instruktioner.

Det är svårt att hitta ett bra sätt att beskriva vad programmering är och vad det kan handla om, men i vår workshop kommer vi att arbeta med att ge våra robotar ett antal instruktioner som den sen kommer att utföra i tur och ordning. Detta kallas ibland för sekvensprogrammering, och speglar i grunden hur en dator arbetar. I dagens datorer utförs dock ofta flera processer samtidigt, datorn arbetar då parallellt. Vi kommer också att införa ett antal villkor för robotens beteende. På så vis kan vi använda de sensorer som finns på robotens styrkort.

Robotens instruktioner – Algoritmer

Programmering är till mångt och mycket samma sak som problemlösning där man försöker att bryta ner en stor uppgift till flera, mycket mindre instruktioner som roboten eller datorn kan arbeta med. En begränsad mängd instruktioner – med en start- och en slutpunkt – brukar kallas för en algoritm. De enklaste algoritmerna skulle kunna jämföras med till exempel ett recept för att baka, medan de klurigare algoritmerna kan liknas vid små beskrivningar över exempelvis en robots beteende i olika typer av miljöer och händelser.

För att förstå hur en algoritm fungerar kan du testa att beskriva något vardagligt, till exempel att tvätta händerna eller att dricka ett glas vatten. Om du lyckas dela upp detta i avgränsade delar så har du lyckats att göra en algoritm. Varje del av din beskrivning kan sannolikt brytas ner i ytterligare mindre bitar och kombineras med andra algoritmer. Programmering handlar alltså till stora delar om att, med hjälp av ett språk kunna beskriva hur exempelvis en robot ska bete sig utifrån de olika situationer den kan befinna sig i.

Programmering, att lära sig nya språk

Att prata med robotar

Idag går det att prata ganska bra med robotar, och de förstår ganska mycket av vad vi säger. Men givetvis kan inte en robot förstå allt vi säger och inte heller hur man säger det. En robot har dessutom svårt att förstå hur vi känner oss, vad som gör oss glada och så vidare. Tekniken för att känna igen exempelvis känslor blir hela tiden bättre och bättre, men en dator kan inte "förstå" det som inte går att mäta och omvandlas till tydliga elektriska signaler.

Många gånger styr vi våra robotar med påhittade språk, knappar eller symboler på en skärm. I vissa fall kan vi också programmera robotar genom att helt enkelt flytta deras armar till olika positioner och spara dessa i ett minne. Det finns alltså flera olika sätt att programmera, alla med sina för- och nackdelar.

Att tänka i algoritmer

För att roboten ska förstå vad du vill att den ska göra måste du alltså kunna beskriva det du vill att den ska göra i små och stegvisa instruktioner, du behöver också veta vilka instruktioner som du kan få roboten kan reagera på. Ytterligare en sak att fundera över är vilken typ av information som roboten kan känna av och vilken information du måste ge den. Sen är det bara att köra.

Sensorer

Den typ av robotar som vi kommer att arbeta med på Visualiseringscenter är utrustade med ett flertal sensorer. De olika sensorerna ger möjligheten att programmera en robot som reagerar på sin omgivning, men det finns också sensorer som håller koll på robotens rörelser. På så vis kan programmeraren skriva algoritmer som fyller roboten med en artificiell intelligens. På så vis kan roboten lösa olika typer av problem eller leverera tjänster utan att någon behöver programmera om den hela tiden.

Sensorerna fungerar på olika sätt, men har det gemensamt att de gör olika mätningar av miljön. Dessa mätningar omvandlas sedan till digital form som styrsystemet i roboten kan behandla.

A/D-omvandling

A/D-omvandling står för analog-till-digital-omvandling; alltså att översätta uppmätt information till en signal som en dator kan arbeta med. Något förenklat fungerar processen så här:

1. Signalen från sensorn mäts upp med jämna mellanrum. Dessa mätningar omvandlas sedan och bearbetas i A/D-omvandlaren. Hur ofta inläsningen sker kallar man för samplingsfrekvens, och den mäts i Hertz (Hz). Vill man veta hur långt det är mellan varje sampling så är det bara att ta siffran ett och dela den med samplingsfrekvensen. Det samma gäller tvärt om; siffran ett delat med Tidsintervallet ger samplingsfrekvensen.

2. Den uppmätta (samplade) signalen jämförs sedan med ett antal förutbestämda spänningsnivåer. Hur många nivåer som används bestäms av hur många bitar – ettor och nollor – som omvandlaren använder sig av. Detta brukar kallas för bit-djup eller upplösning. Om A/D-omvandlaren arbetar med åtta bitar finns det 256 olika värden att jämföra med, 10 bitar ger 1024 värden och så vidare. Varje ny bit fördubblar alltså antalet möjliga nivåer.

3. Den signal som kom in i omvandlaren kan nu omvandlas till en digital signal, beskriven med ettor och nollor; en signal som styrkortet och processorn kan börja arbeta med. Hur omvandlingen fungerar är olika i olika modeller och applikationer. Vissa tekniker är snabbare än andra, vissa tekniker kräver mer datorkraft än andra, och så vidare. Det är helt enkelt upp till konstruktörerna att välja det bästa sättet att omvandla signalerna.

När styrsystemet väl har fått en digital signal kan det börja reagera utefter den. Hur systemet ska reagera har programmeraren förutbestämt.

D/A-omvandling

För att till exempel en robot ska kunna svänga av vid rätt tillfälle, blinka med lampor eller gripa tag i något krävs att styrsystemet kan skicka ut egna elektriska signaler. Detta kallas för en digital-till-analog-omvandling. Denna omvandling är minst lika viktig om exempelvis en motor ska kunna styras exakt. Processen går till ungefär så här:

1. Robotens styrprogram behandlar signalerna som kommer in från sensorerna. Ut ur programmet kommer ett eller flera digitala värden i form av ettor och nollor.

2. Med jämna mellanrum jämförs de digitala värdena med ett antal spänningsnivåer. Hur noggrant man kan styra roboten beror på hur många bitar som D/A-omvandlaren arbetar med. Ju fler bitar, desto fler nivåer att arbeta med.

3. En styrström skickas sedan till exempelvis en motor, en lampa eller en display. På så vis kan styrsystemet bestämma hur en robot ska bete sig.

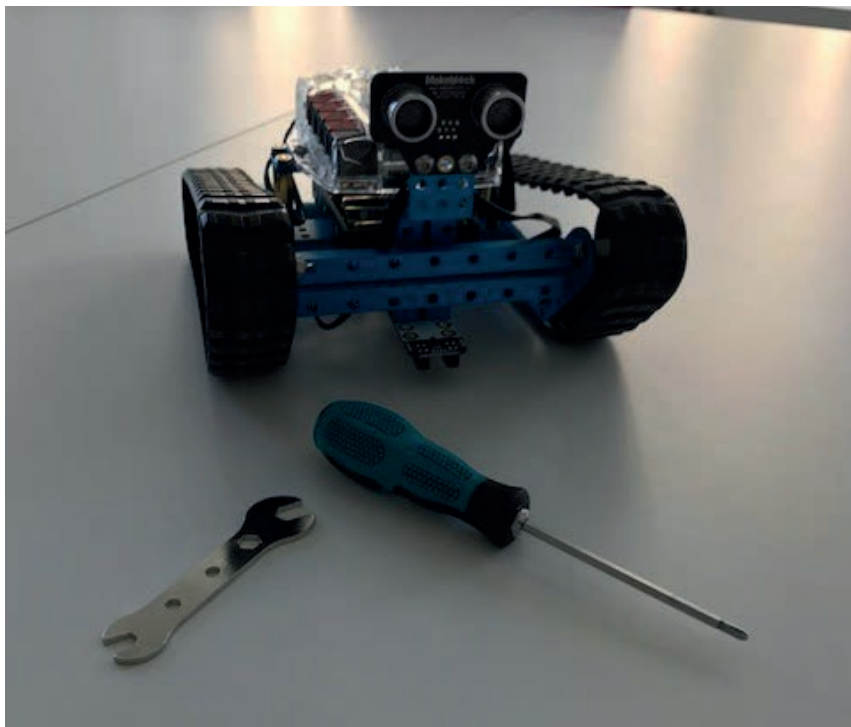
Olika nivåer av programmering

För att ett styrsystem ska fungera så måste alla olika delar kunna samarbeta på ett bra sätt utan att användaren ska behöva fundera så mycket över alla små detaljer. För att få en bra förståelse för hur allt fungerar kan det vara en bra idé att prata om programmering i olika nivåer. Den låga nivån ligger nära det som sker på styrkortet och de höga nivåerna sker där användaren finns.

Varje nivå kräver olika kunskaper hos programmeraren, och mellan varje nivå krävs en gränssyta för att information och signaler ska kunna "översättas" på ett bra vis. Här pratar man ofta om programmeringsspråk och utvecklingsmiljöer; olika vis att kommunicera med tekniken. En kommunikation som så småningom ska landa i ett språk som vi människor kan förstå. Under workshoppen kommer ni att jobba med en grafisk miljö där kommunikationen med roboten styrs med hjälp av olika block. Detta är ett typiskt exempel på högnivåprogrammering.

Programmeringsspråken är givetvis inte samma sak som våra talade och skrivna språk, men faktum är att de också har en speciell grammatik att följa och skrivregler som är nödvändiga för att styrsystemet ska "förstå" vad vi vill säga till den. Det som skiljer ett programmeringsspråk från talade språk är att vi människor kan prata ganska så felaktigt, men ändå bli förstådda. Talade språk utvecklas också ganska spontant och det förmedlar väldigt mycket information om sådant vi inte riktigt kan prata om. Andra saker som skiljer programmeringsspråk från vårt naturliga språk är att programmeringsspråken inte kräver att vi har någon att prata med och att det kan byggas upp från grunden. Att göra samma sak med ett talat språk skulle vara ytterst svårt – eller kanske rent av omöjligt.

Workshop – Beskrivning



Våra workshoppar kommer att köras med en mBot Ranger. Varje grupp kommer att arbeta med en robot för att lösa ett antal uppdrag. Robotarna kopplas, via USB till en laptop. Rangern går att programmera och köras frikopplad från datorn, men den går även att köra med USB-kabeln inkopplad. Det krävs en del försiktighet för att den inte ska dra iväg för snabbt med hela datorn i släptåg.

mBlock

Den mjukvara vi kommer att jobba med kallas för mBlock. Det är ett gränssnitt som tillåter eleverna att programmera roboten med olika fördefinierade block, samtidigt som de också ser den programkod som roboten arbetar med. Det grafiska gränssnittet gör att vi kommer åt en del funktioner på ett enkelt vis.

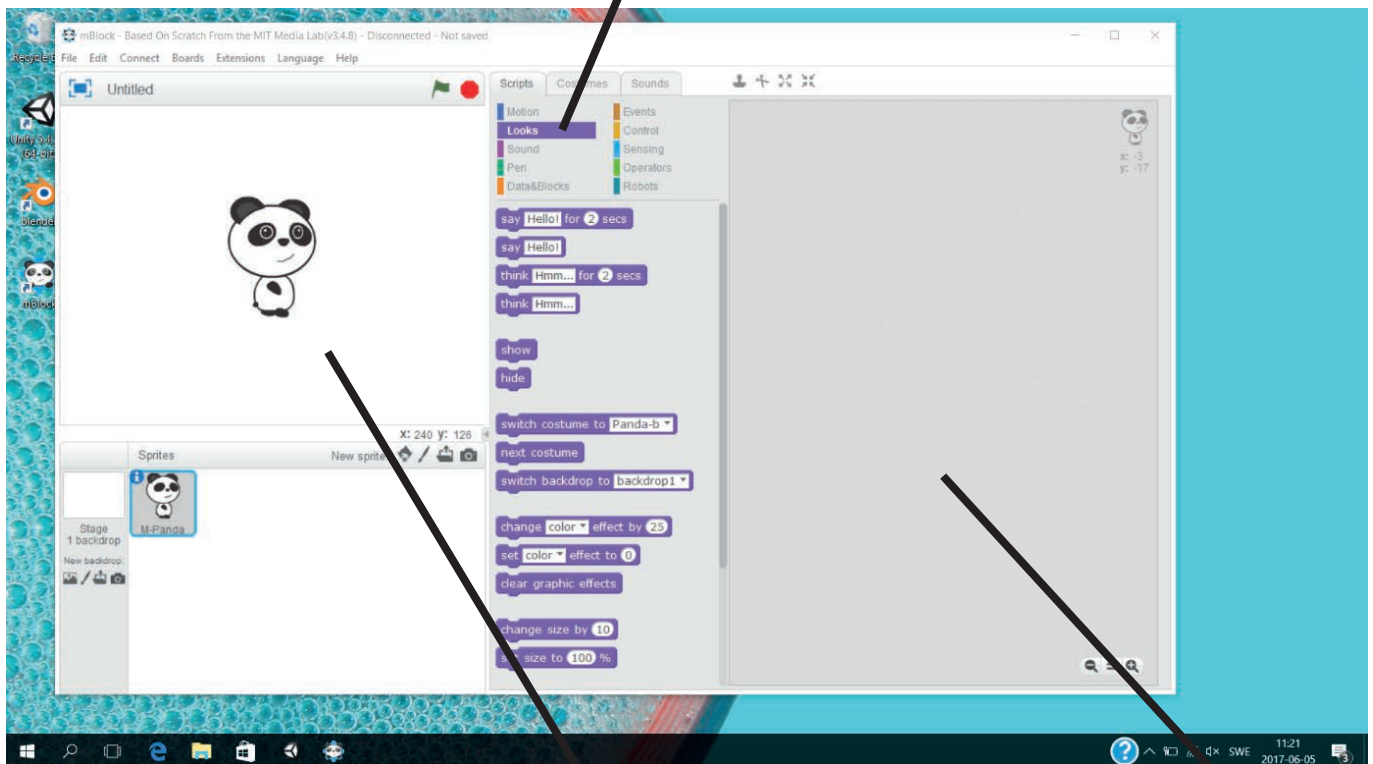
För att kunna hantera mBlock krävs inga större förkunskaper, men det är bra om ni har testat att programmera med Scratch (<https://scratch.mit.edu/>) samt att ni har tittat över manualerna till programvaran (<http://www.mblock.cc/>).

Uppstart och uppkoppling av utrustningen...

... men först några inledande ord om vad som finns på skärmen

Funktioner och Operatörer: Här hämtas (och definieras) de byggblock och variabler som krävs för att programmeringen ska flyta. Här finns alla bitar som hanterar exempelvis loopar, jämförelser, villkor, etc.

Ingredienserna till algoritmerna

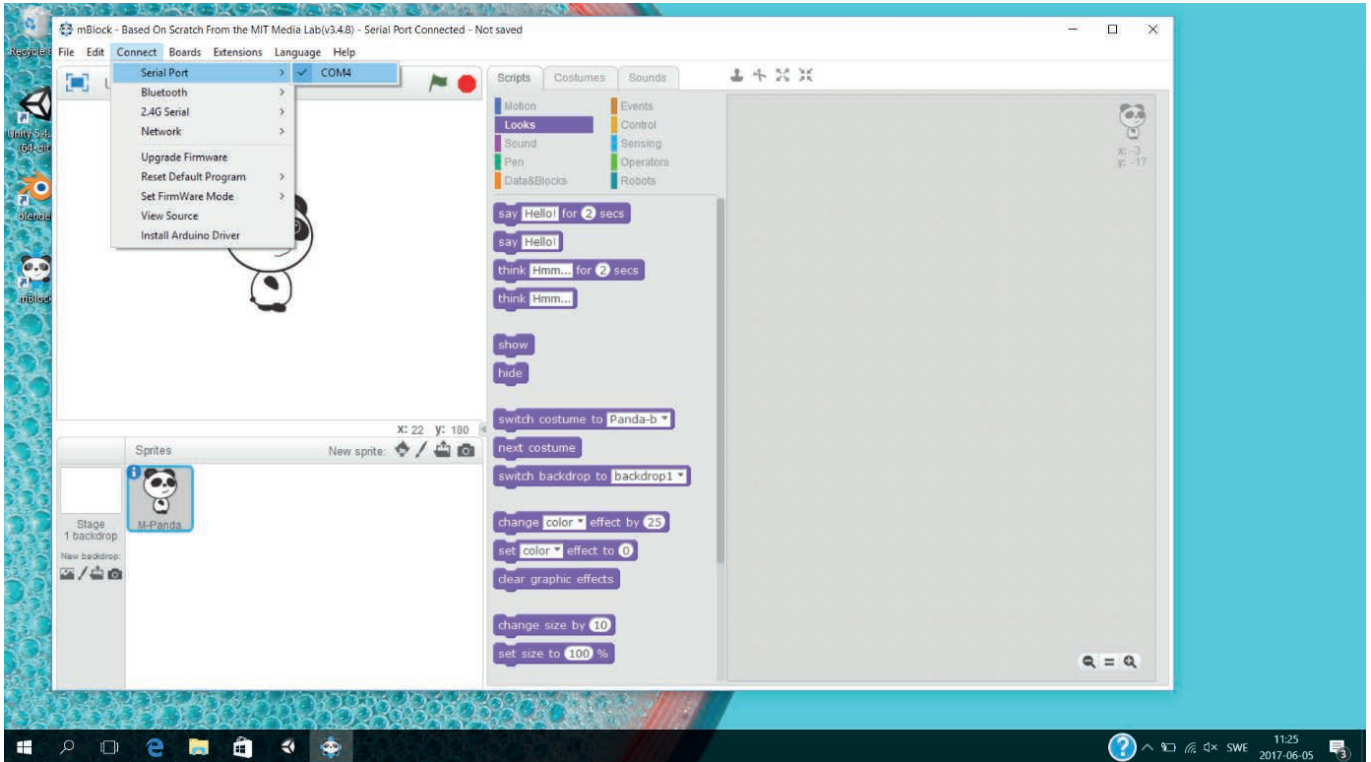


Sprite: En liten figur som styrs av de olika blocken. Spriten kan se olika ut, och den kan utföra i stort sett samma sak som roboten. Med andra ord så kan den användas som felsökningsverktyg eller som en del i en introduktion till mjukvaran.

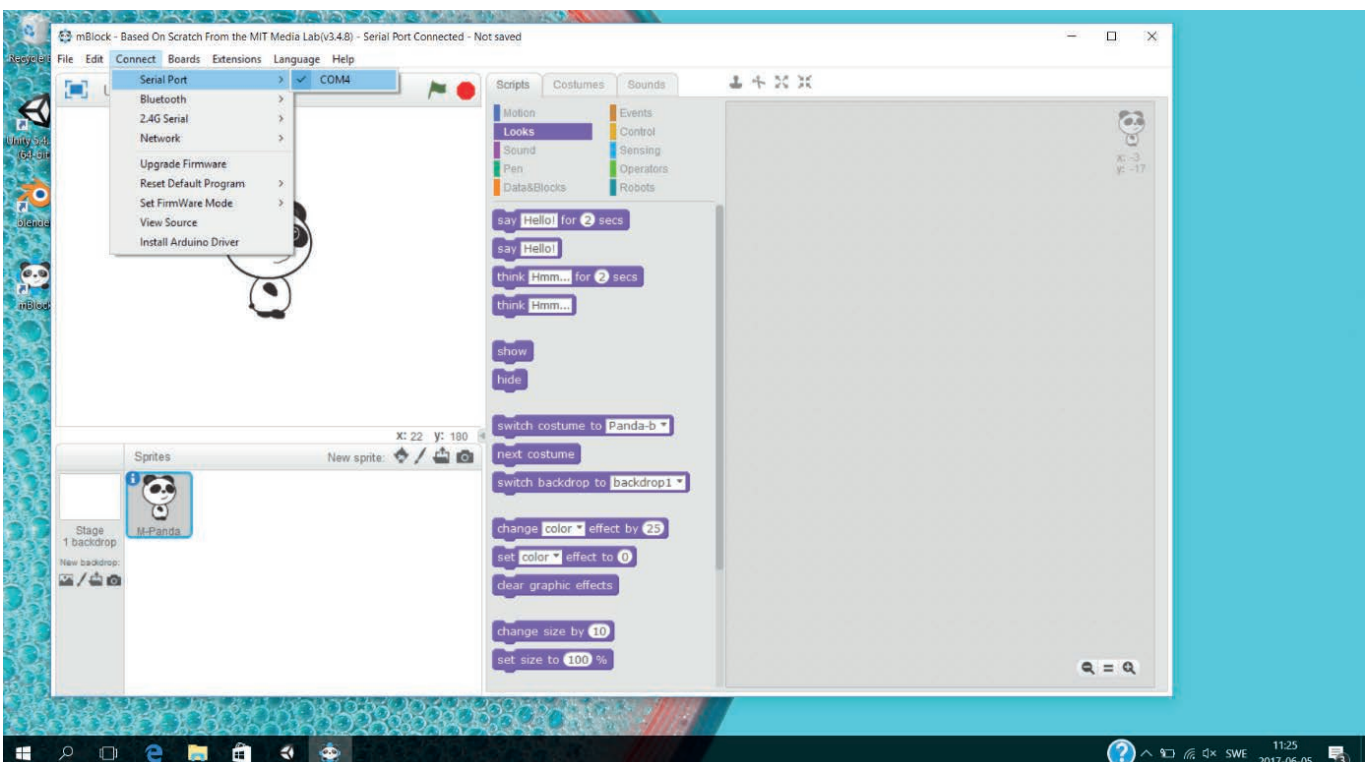
Arbetsyta: Den delen av ytan där alla kodblock dras ut till. Kodblocken kan här konstrueras i mindre delar, men körs hela tiden sekventiellt. Här finns alltså programmets algoritmer

Koppling mellan dator och robot

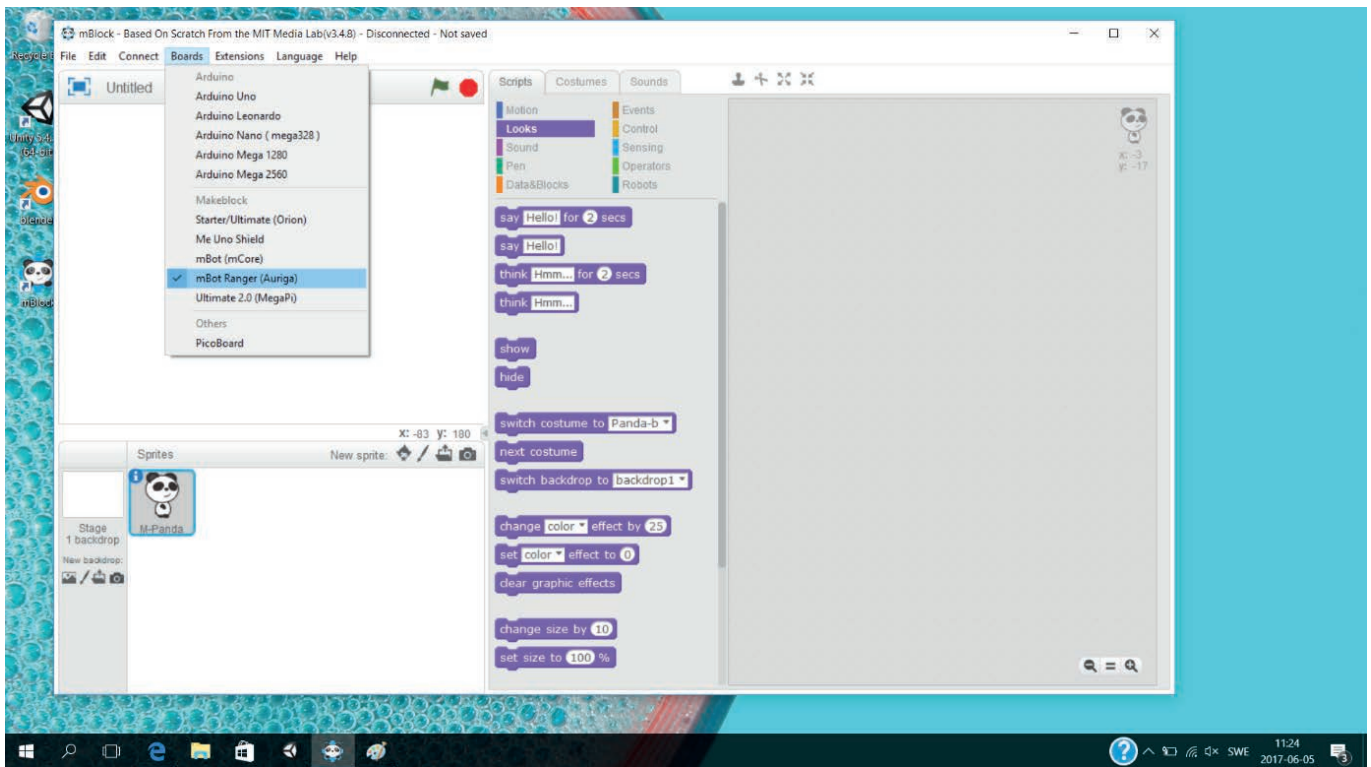
1. Börja med att koppla samman roboten och datorns. Allt utrustning borde hitta varandra automatiskt, men det är inte säkert att så sker.



2. Ställ in mjukvaran för att arbeta med mBot Ranger. Uppdatera sedan firmware samt återställ roboten till ursprungsprogrammet. (Finns under 'Connect'-menyn.)

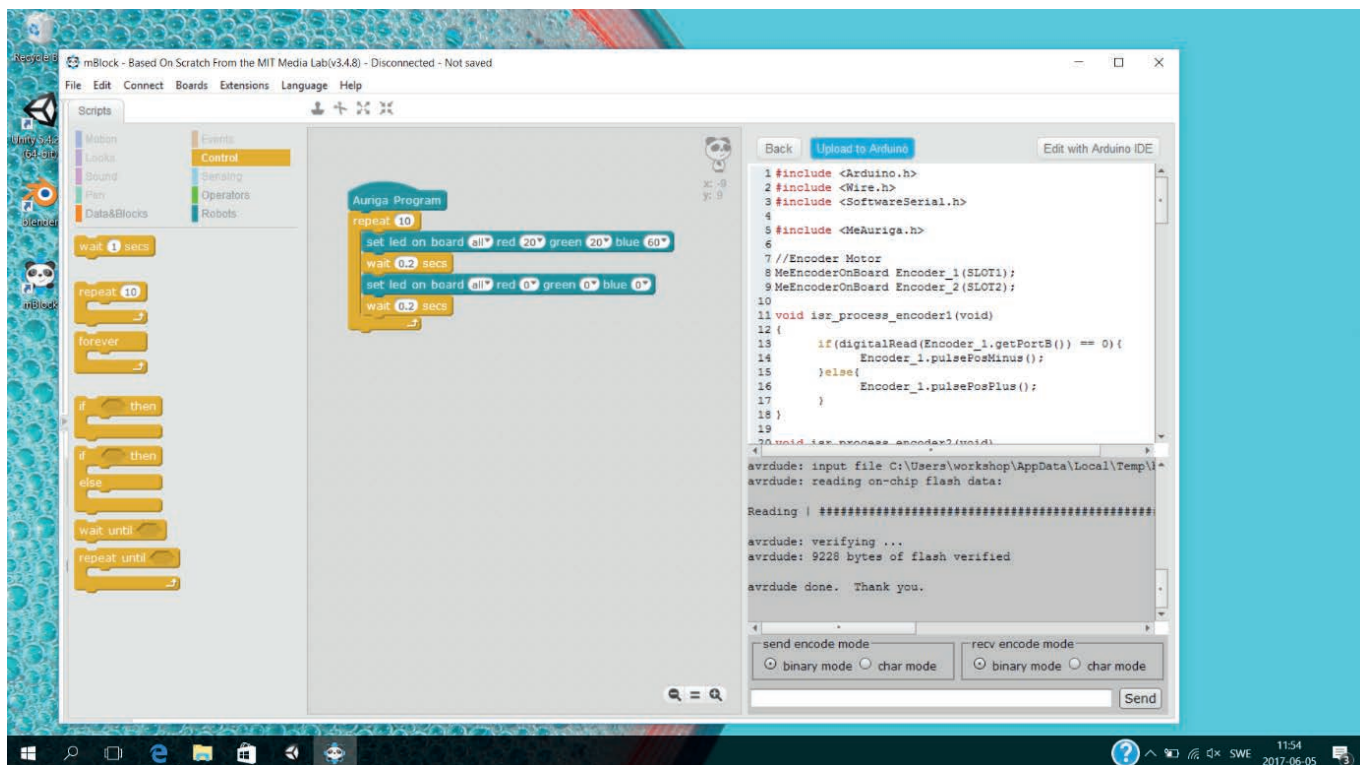


3. Ställ om mjukvaran till 'Arduino mode'



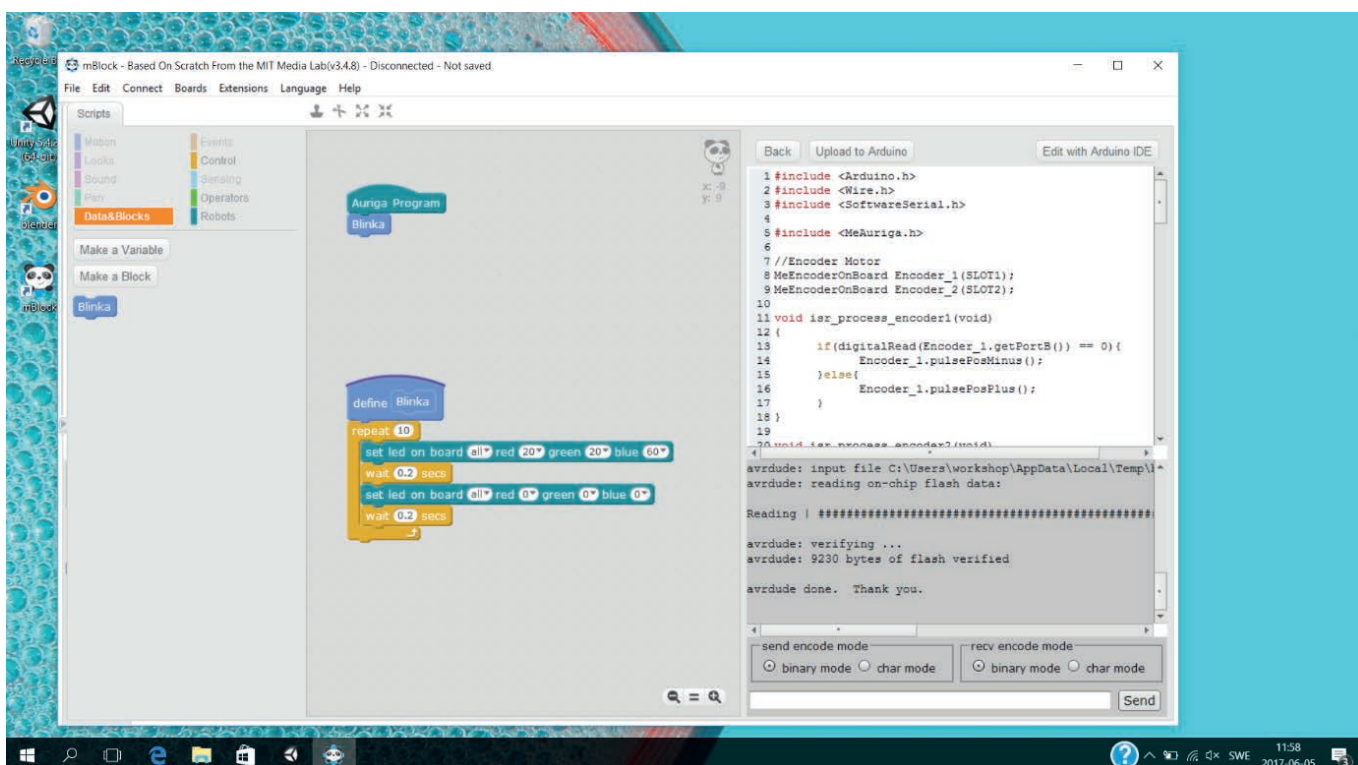
Ett första exempel - Få LED-lamporna att blinka

1. Skapa en enkel algoritm för att tända och släcka samtliga LED-lampor



När algoritmen är klar så klickar du på 'Upload to Arduino' och kopplar loss roboten ur datorn.

2. Algoritmen kan sen göras om till ett data-block som man sedan har tillgång till på fler ställen i sitt program.



3. Sen är det bara att hitta nya block att jobba med

