# Lektion | Arbetsblad |.| - Möt Edison

Edison är en liten LEGOanpassad och programmerbar robot.

Edison kan kommunicera med omvärlden med hjälp av sina sensorer. I bilden finns Edisons sensorer, knappar och strömbrytare utmärkta.



Lär känna Edisons sensorer och knappar

Startknapp – Startar ett program Stoppknapp – Tryck för att avsluta ett program Inspelningsknapp – 1 tryck = ladda ner ett program, 3 tryck = läs streckkod



Edisons strömknapp och sensor för linjeigenkänning

EdComm-kabeln används för att ladda ner dina program till Edison. Kabeln kopplas till uttaget för hörlurar på din dator eller tablett.

Edisons sensor för att följa en linje består av två delar, en röd LED-lampa och en ljussensor.

Sensorn kan också läsa speciella streckkoder som aktiverar förinstallerade program i roboten.



EdComm programmeringskabel

# Lektion I Arbetsblad I.2 – Programmering med streckkoder

### Läs av streckkoden

- 1. Placera Edison på höger sida om streckkoden så att han är riktad mot koden.
- 2. Tryck 3 gånger på inspelningsknappen
- 3. Edison kommer att köra framåt för att läsa av koden



# Lektion | Arbetsblad |.3 - Möt EdWare

Det här är EdWare, mjukvaran som behövs för att programmera Edison.



För att börja programmera skall du klicka på ikoner i paletten och dra ut dem på programmeringsfältet. Placera ikonerna mellan "start"- och "end"-ikonerna. Välj en ikon och ändra inställningarna i egenskapsfältet. Så bestämmer du hur Edison skall reagera på just den ikonen. Använd hjälptexten som guide medan du programmerar. Allt du behöver veta om en ikon hittar du där. I fältet för variabler kan du skapa variabler och kontrollera små bitar av Edisons minne. Det lär du dig mera om senare.

#### Vad heter de fyra huvudgrupperna av ikoner?

# Lektion I Arbetsblad I.4 - Ladda ner ett testprogram

Öppna testprogrammet. Filnamn: Test Program.edw (i Windows: C:\Program Files\EdWare\My Programs)



# Ett program ser ut så här. Edison ser på en ikon i taget och gör vad ikonen säger. Pilarna visar i vilken ordning Edison läser ikonerna. I det här programmet finns det en *slinga*. Den gör att programmet upprepas gång på gång ända tills batteriet tar slut (eller användaren avslutar programmet med stoppknappen).

För att ladda ner programmet skall ni koppla in EdComm-kabeln till uttaget för hörlurar på datorn/tabletten och skruva upp volymen till full styrka. Koppla andra änden av kabeln till Edison så som bilden visar.



Gör följande steg för att ladda ner programmet:

- 1. Tryck på Edisons inspelningsknapp (den runda) en gång
- 2. Tryck på "Program Edison" och klicka sedan på "Start Download".
- 3. Tryck på Edisons startknapp för att starta programmet

#### Vad gjorde roboten då du tryckte på startknappen?

Vilka ikoner bestämmer hur Edison rör sig? Skriv ner hur du tänkte.

Hur kom programmet från datorn till roboten?

www.meetedison.com

www.roboticswps.com.au

# Lektion 2 Arbetsblad 2.1 – Kör roboten framåt

Skriv ett program som kör Edison framåt. Använd dig av aktivitetsbladet eller färgat tejp på bordet som start- och stopplinjer.

Dual Drive P	Constant	,	Variable		
Direction: F	orward	▼ <-Const	tant	-	
Speed: 5	•	<-Const	tant	•	
	•				
► start			X		End
► start	dua		event wait		∎ ∈nd
<b>Start</b>	Uait Properti	ies Wait Until:	wait		∎ ∈nd
Start	Wait Properti	ies Wait Until: ?	vent wait	ant	end

Ställ in "Sekunder passerar" (i stället för ?) med en tid i sekunder. Roboten kör framåt detta antal sekunder. Ställ roboten bakom startlinjen och se om den stannar före mållinjen.

Den kortaste tiden man kan ställa in är 0,01 sekunder.

Den längsta tiden man kan ställa in är 327,67 sekunder.

Testa olika tider tills din robot startar bakom startlinjen och kör framåt exakt rätt antal sekunder så att den stannar just innan mållinjen.

Vad är den korrekta tiden som får din robot att köra från startpunkten till mållinjen?

Beskriv vad roboten gör och varför.

www.meetedison.com

# Lektion 2 Arbetsblad 2.2 – Kör roboten bakåt

Skriv ett program som kör Edison bakåt. Använd dig av aktivitetsbladet eller färgat tejp på bordet som start- och stopplinjer.

buurbhire	Constant	Va	ariable		
Direction:	Backward	▼ <-Consta	nt	-	
Speed:	5 🔻	<-Consta	nt	•	
	•	,			
► start			vent wait		€nd
► start	ent Wait Propert	ies Wait Until:	event wait		€nd
start	ent Wait Propert Seconds pass	ies Wait Until:	vent wait	ant	€nd

Ställ in "Sekunder passerar" (i stället för ?) med en tid i sekunder för att låta roboten köra framåt detta antal sekunder. Ställ roboten bakom startlinjen och se om den stannar före mållinjen.

Den kortaste tiden man kan ställa in är 0,01 sekunder.

Den längsta tiden man kan ställa in är 327,67 sekunder.

Testa olika tider tills din robot startar bakom startlinjen och kör bakåt exakt rätt antal sekunder så att den stannar just innan mållinjen.

Vad är den korrekta tiden som får din robot att köra från startpunkten till mållinjen?

Beskriv vad roboten gör och varför.

# Lektion 2 Arbetsblad 2.3 – Framåt och sedan bakåt

Skriv ett program som kör Edison först framåt och sedan bakåt. Använd dig av aktivitetsbladet eller färgat tejp på bordet som start- och stopplinjer.



Ställ in "Sekunder passerar" (i stället för ?) med en tid i sekunder på båda ställena. Ställ roboten bakom startlinjen och se om den stannar före mållinjen. Kontrollera också att den stannar före startlinjen när den backat klart.

Den kortaste tiden man kan ställa in är 0,01 sekunder.

Den längsta tiden man kan ställa in är 327,67 sekunder.

Testa olika tider tills din robot startar bakom startlinjen och kör framåt exakt rätt antal sekunder. Roboten skall stanna just innan mållinjen och sedan köra bakåt så att den stannar just innan startlinjen.

Vilka är de korrekta tiderna som roboten skall köra framåt och sedan bakåt?

Framåt \_\_\_\_\_ Bakåt \_\_\_\_\_

Beskriv vad roboten gör och varför.

### Lektion 2 Arbetsblad 2.4 - Hastigheter

Skriv ett program som kör Edison först framåt och sedan bakåt. Använd dig av aktivitetsbladet eller färgat tejp på bordet som start- och stopplinjer.

Speed: Spe	Dual Drive Properties Constant Variable Direction: Forward C-Constant	Dual Drive Properties Constant Variable Direction: Backward  <-Constant
wait Until:       Event Wait Properties         Wait Until:       Wait Until:         Seconds pass          Event happens       Kevpad         Triangle button pressed	Speed: ?	Speed: Constant Co
Seconds pass ? <-Constant    Seconds pass ? <	rent Wait Properties Wait Until:	Event Wait Properties Wait Until:
	Seconds pass ? <-Constant	Seconds pass                    Event happens              Kevpad               Triangle button pressed

Denna gång kan du ändra både tiden och robotens hastighet!

Känn dig fri att lägga till andra ikoner, sådana som gör att roboten blinkar med LEDlamporna, piper eller till och med spelar musik.

1. Starta bakom startlinjen och stanna före mållinjen, kör sedan bakåt och stanna före startlinjen. Gör detta så snabbt som möjligt!

#### Vilka är inställningarna för din snabbaste körrunda?

Hastighet framåt \_\_\_\_\_ Tid framåt \_\_\_\_\_

Hastighet bakåt \_\_\_\_\_ Tid bakåt \_\_\_\_\_

2. Starta bakom startlinjen och stanna före mållinjen, kör sedan bakåt och stanna före startlinjen. Gör detta så långsamt som möjligt!

Vilka är inställningarna för din långsammaste körrunda?

Hastighet framåt \_\_\_\_\_ Tid framåt \_\_\_\_\_

Hastighet bakåt \_\_\_\_\_ Tid bakåt \_\_\_\_\_

### Lektion 2 Aktivitetsblad 2.1





# STARTLINJE

# Lektion 3 Arbetsblad 3.1 – Sväng 90° åt höger

Skriv följande program för att svänga Edison 90 grader (90°) åt höger. Använd aktivitetsblad 3.1 eller markerade linjer på bordet som start- och stoppmärken.

Dual Drive Prope C	rties onstant	Variable		
Direction: Spin r	ight ▼ <-C	onstant	-	
Speed: 5 🔻	· <-C	onstant	<b>•</b>	
start	dual drive		end	
1	Event Wait Propert	ies Wait Until:		
0	Seconds pass	?	<-Constant	•
	Event happens	Keypad 🔻	Triangle button pres	ssed 👻

Ställ in "Sekunder passerar" (i stället för ?) med en tid i sekunder för att få roboten att svänga höger exakt 90°.

Den kortaste tiden man kan ställa in är 0,01 sekunder.

Den längsta tiden man kan ställa in är 327,67 sekunder.

Testa olika tider tills din robot svänger höger från startmärket exakt tills den når stoppmärket.

Vad är den korrekta tiden för att få din robot att svänga sig 90°?

Beskriv vad roboten gör och varför.

# Lektion 3 Arbetsblad 3.2 – Sväng 180° åt vänster

Skriv följande program för att svänga Edison 180 grader (180°) åt vänster. Använd aktivitetsblad 3.1 eller markerade linjer på bordet som start- och stoppmärken.

Dual Drive Pro	perties Constant	Variable	
Direction: Spi	n left 🔹 🔻	<-Constant	-
Speed: 5	•	<-Constant	-
<b>b</b> start	dual		end
	Event Wait Pro	perties Wait Until:	
	Seconds participation	ss ?	<-Constant
	Event happe	ens Keypad 🔻	Triangle button pressed 👻

Ställ in "Sekunder passerar" (i stället för ?) med en tid i sekunder för att få roboten att svänga vänster exakt 180°.

Den kortaste tiden man kan ställa in är 0,01 sekunder.

Den längsta tiden man kan ställa in är 327,67 sekunder.

Testa olika tider tills din robot svänger vänster från startmärket exakt tills den når stoppmärket.

Vad är den korrekta tiden för att få din robot att svänga sig 180°?

Beskriv vad roboten gör och varför.

# Lektion 3 Arbetsblad 3.3 – Sväng höger, sedan vänster

Skriv följande program. Programmet svänger Edison först 90 grader (90°) höger och sedan 180 grader (180°) åt vänster. Använd aktivitetsblad 3.1 eller markerade linjer på bordet som start- och stoppmärken.



Ställ in "Sekunder passerar" (i stället för ?) med tider i sekunder för att få roboten att först svänga höger 90° och sedan vänster exakt 270°.

Den kortaste tiden man kan ställa in är 0,01 sekunder.

Den längsta tiden man kan ställa in är 327,67 sekunder.

Testa olika tider tills din robot svänger höger från startmärket exakt tills den når stoppmärket och sedan svänger vänster tills den nåt stoppmärket längst ner på sidan.

Vad är den korrekta tiden för att få din robot att först svänga 90°, sedan 270°?

90°: \_\_\_\_\_, 270°: \_\_\_\_\_

Beskriv vad roboten gör och varför?

### Lektion 3 Arbetsblad 3.4 – Minibana

Använd de kunskaper i programmering som du har för att köra Edison genom banan på aktivitetsblad 3.2.

Roboten måste starta framför startlinjen och stanna efter mållinjen och den får inte köra på kantlinjerna.

Du kommer att behöva kombinera många olika programmeringsikoner för att köra roboten framåt och göra de svängar som krävs.

Tips: Framåt, vänster, framåt, höger, framåt

Beskriv vad roboten gör och varför?

#### Vilka utmaningar stötte du på när du försökte köra Edison genom banan?

#### **Avancerad tävling:**

Vem kan köra snabbast genom banan?

Det finns inget behov av stoppur. Addera bara tiderna från alla vänteikoner.

Kom ihåg: Roboten som vinner måste starta framför startlinjen och stanna först efter mållinjen och får inte köra på linjerna.

Hur länge tar det för din robot att köra genom labyrinten?

Namn:

### Lektion 3 Aktivitetsblad 3.1 - Sväng!

Placera roboten i det utmärkta området och starta sedan ditt program. Din robot borde:

Program 1 – Svänga höger från start 0° till stopp 90°

Program 2 – Svänga vänster från start 0° till stopp 180°

Program 3 – Svänga höger från start 0° till stopp 90°, sedan svänga vänster 270° till stopp 180°



### Lektion 3 Aktivitetsblad 3.2 – Minibana



### Lektion 4 Arbetsblad 4.1 – Utmaning

Välj en utmaning för dig och din robot. Programmera Edison så att han klarar av att utföra den utmaning du valt.

Här är några exempel, men du kan hitta på en egen utmaning.

- Kör runt ett hinder, ex. en kopp eller en penal
- Kör runt kanten på ett bord utan att köra över kanten
- Skapa en labyrint på ett stort papper
- Skapa en labyrint av en byggsats som exempelvis LEGO

Kom ihåg att du kan använda andra programmeringsikoner i ditt program, ex. blinkande LED-lampor, ljud eller musik!

Vad valde du för utmaning för din robot?

#### Vad var svårt med att skriva detta program?

Vilka andra programmeringsikoner använde du och vad gjorde de?

Ν	la	m	n	1
---	----	---	---	---

### Lektion 4 Arbetsblad 4.2 – Mexikanska vågen

Det här är en rolig aktivitet för hela klassen. Allas robotar kör samma program men de startar inte samtidigt. Resultatet liknar en mexikans våg eller en koreografisk dans.

### Slingor

Använd en slinga i ditt program för att få rörelserna att upprepa sig.



### Vad du behöver göra:

Skriv ett kort program med en "cool" sekvens av robotrörelser. Glöm inte att använda dig av ljus och ljud i ditt program.

När du skrivit din sekvens och testat den, låt klassen eller läraren utvärdera och välja det "roligaste" programmet i klassen.

Om ditt blir valt, dela ditt program med resten av klassen. Om någon annans program blir valt, öppna deras program på din dator och lägg till en vänteikon i början av programmet. Ställ in tiden enligt lärarens instruktioner.

 Tid i vänteikonen:
 Robotnummer:
 (för att ställa robotarna på rad)

Placera alla robotar på en rad i samma ordning som tidsfördröjningarna. Ställ den med kortast fördröjningstid i början och den med längst i slutet. Sedan trycket ni alla samtidigt på startknappen.

Beundra robotarnas uppvisning.

Beskriv din robots coola rörelser.

www.meetedison.com

www.roboticswps.com.au

sida 35

inamn:	Ν	а	m	n	i
--------	---	---	---	---	---

### Lektion 5 Design | Arbetsblad 5. | – Mitt program

I framtiden kommer robotar att vara våra hjälpredor. Vi har redan dammsugaren Roomba, så framtiden är inte långt härifrån. Tänk på en uppgift som din robot kan utföra. Skriv sedan ett program så att Edison kan utföra denna uppgift.

Här är några exempel:

- Dansa till musik en underhållande robot
- Dammsugare mycket användbar i hemmet
- Vaktrobot vaktar din penal

### I. Diskutera

Diskutera med ditt par eller din grupp och bestäm er för en användbar uppgift som roboten kan utföra. Välj något ni känner att ni kan programmera.

### 2. Beskriv

Innan du börjar skriva ditt program får du beskriva problemet som din robot skall lösa och hur den skall lösa det.

#### Problemet är...\_\_\_\_

Min robot löser detta genom att...\_\_\_\_

### 3. Skriv ditt program

Planera ditt program före du börjar lägga in ikoner. Använd dig av de ikoner du lärt dig använda, men känn dig ändå fri att pröva på nya ikoner.

#### 4. Misslyckande?

Allting går inte som man tänkt sig den första gången. Ingen panik! Försök igen, men beskriv först vad som gick fel. Varför fungerade inte ditt program den första gången? Kom ihåg att Thomas Edison misslyckades 10 000 gånger innan han lyckades med att uppfinna sin berömda gödlampa. Så, *försök igen!* 

### Lektion 5 Design 1 Arbetsblad 5.2 - Mitt program

### 5. Beskriv några av de ikoner som används i ditt program

Rita och färglägg programmeringsikonerna nedan. Beskriv sedan vad de gör i ditt program.

Vad heter denna ikon?
Vad gör denna ikon?
Vad heter denna ikon?
Vad gör denna ikon?
•
Vad heter denna ikon?
Vad gör denna ikon?
J

# Lektion 6 Arbetsblad 6. I – Blinka med LED som reaktion på en klapp

Skriv ett program som får Edison att blinka med sin vänstra LED-lampa som reaktion på ett högt ljud. Ett högt ljud kan vara exempelvis en klapp.



I detta program väntar inte den första vänteikonen på att en tid har gått. I stället väntar programmet på att en specifik händelse har skett. Först då kommer programmet att fortsätta.

Den första vänteikonen behöver ställas in. Välj "Händelse inträffar" (Event happens) och välj sedan den händelse som du tror behövs för att reagera på en klapp.

### På vilket avstånd detekterar roboten din klapp?

Vad spelar slingan för roll i programmet och vad skulle hända om slingan inte skulle vara med?

# Lektion 6 Arbetsblad 6.2 – Kör som reaktion på en klapp



Skriv följande program som får Edison att köra framåt som reaktion på en klapp.

Edisonrobotens ljudsensorer är känsliga. De reagerar inte bara på klappar, utan också på andra höga ljud. Eftersom robotens motorer, växlar och hjul också ger ljud ifrån sig kan sensorerna reagera på detta. För att förhindra att roboten triggas av dessa ljud finns det två nya vänteikoner i programmet. De är markerade med orangea stjärnor. Första vänteikonen är inställd på 0,1 sekunder och ger robotens motorer tid att stanna. Den andra vänteikonen är inställd på att vänta på ett ljud. Ikonen kommer att få programmet att fortsätta eftersom den kommer att reagera på ljudet av motorerna.

Du måste lägga till de här ikonerna efter att motorerna har stannat om du vill använda sensorn för att detektera klappar på nytt. Annars tar programmet slut och kommer inte vidare.

Experimentera med olika riktningar i den första körikonen och med olika tider i den följande vänteikonen för att skapa olika sorters rörelser.

#### Vilka andra riktningar och tider prövade du på?

Experiment 1: riktning \_\_\_\_\_, tid\_\_\_\_\_

Experiment 2: riktning \_\_\_\_\_, tid\_\_\_\_\_

Experiment 3: riktning \_\_\_\_\_, tid\_\_\_\_\_

www.meetedison.com

www.roboticswps.com.au

# Lektion 6 Arbetsblad 6.3 – Dansa som reaktion på en klapp

Med hjälp av det du lärt dig de senaste två övningar kan du nu skapa en dansrutin. Den här gången dansar roboten som reaktion på dina klappar.



Du kommer att behöva minst två olika dansrörelser, men du kan lägga till så många du vill. Programmet ovan har två manövrar med ett par rörelser i var. De här två upprepar sig eftersom de ligger i en slinga.

Du kan också pröva på att lägga till två manövrar per klapp.

#### Alternativ utmaning

Om du inte vill programmera en dans kan du bygga en kort hinderbana som roboten kör igen. Du kan styra Edison med hjälp av klappar.. Edison skall sedan komma tillbaka till dig som reaktion på en sista klapp.

Hur många manövrar har ditt program?

Beskriv din robots dans och körmanövrar.

### Rolig klassaktivitet

Välj den bästa dansen i klassen och låt alla programmera in denna i sin robot. Placera sedan alla robotar på en rad och klappa i takt för att få robotarna att dansa.

inami.	Ν	а	m	۱r	۱:
--------	---	---	---	----	----

### Kalibrera sensorn för objektdetektion

Du kan kontrollera känsligheten på Edisons detektionssystem. Genom att göra sensorn känsligare kan Edison märka objekt på längre håll. Om du gör den mindre känslig kommer han bara att märka hinder som ligger väldigt nära.

### Läs av en streckkod

- 1. Placera Edison riktad mot streckkoden på kodens högra sida
- 2. Tryck 3 gånger på inspelningsknappen (den runda)
- 3. Edison kommer att köra framåt och läsa av streckkoden





Streckkod – Kalibrera sensorn för objektdetektion

### Ställ in den maximala känsligheten

Läs först av ovanstående streckkok och tryck sedan på startknappen (triangeln). Edison är nu i kalibreringsläge. Se till att det inte finns några hinder framför Edison.

Vänstra sidans sensorer kalibreras först.

1. Tryck upprepade gånger på startknappen (detta ökar känsligheten) tills den vänstra LED-lampan flimrar.

2. Tryck upprepade gånger på inspelningsknappen (detta minskar på känsligheten) tills LED-lampan helt slutar flimra.

3. Tryck på stoppknappen för att kalibrera den högra sidan.

4. Tryck upprepade gånger på startknappen tills den högra LED-lampan flimrar. Tryck sedan upprepade gånger på inspelningsknappen tills lampan slutar flimra.

5. Tryck på stoppknappen för att slutföra kalibreringen.

### Anpassad känslighet

Du kan ställa in avståndet på vilket Edison märker hinder genom att ställa ett hinder framför Edison på önskat avstånd och upprepa steg 1 till 5.

# Lektion 7 Arbetsblad 7.1 – Detektera hinder med den infraröda sensorn

Edisonroboten är utrustad med infraröd (IR) "syn". IR är inte synligt för människans ögon, så du kan inte se detta ljus. Ljuset låter Edison "se" i mörkret.

För att Edison skall kunna märka hinder skickar den ut infrarött ljud framför sig. Edison har lampor både på den vänstra och högra sidan. Om ljuset reflekteras från ett hinder, ex. en vägg, så detekteras det reflekterade ljuset av Edisons infraröda detektor. Detektorn ligger längst fram i mitten på roboten.

I bilden finns det ett hinder på Edisons vänstra sida. Det är då bara infrarött ljus från den vänstra lampan som reflekteras. Från signalen Edison får in kan han sluta sig till att det finns ett hinder till vänster men inte till höger.

Utsänt infrarött ljus är utmärkt i rött och reflekterat ljus med blått.



Rita in det utsända och reflekterade infraröda ljuset för detta hinder.



# Lektion 7 Arbetsblad 7.2 - Detektera hinder och stanna

Skriv följande program som får Edison att köra framåt ända tills han träffar på ett hinder.

Detect Obstacle Properties Constant Variable	Event Wait Properties Wait Until: Seconds pass 0 <-Constant
IR Transmitter obstacle detection: On 👻 <-Constant 💌	
start	event wait
Dual Drive Properties Constant Va	riable
Direction: Forward  <-Consta	nt 👻
Speed: 5 -Consta	nt 🔹

Den röda detektionsikonen slår på systemet för objektdetektion och starta de infraröda sändarna.

Körikonen är inställd på hastighet 5 för att roboten skall hinna märka av hinder innan han kolliderar med dem. Om hastigheten är för hög kommer Edison inte att hinna stanna utan kollidera med hindret.

På vilket avstånd kan roboten detektera hinder?

Har du sett denna typ av osynlig detektion tidigare och i så fall var?

Var tror du att denna typ av teknologi för detektion skulle kunna vara användbar?

# Lektion 7 Arbetsblad 7.3 - Att undvika hinder

Skriv följande program. Edison kör framåt tills han ser ett hinder, vänder sig 180° och kör ytterligare 1 sekund bort från hindret.



Under lektion 3 (Arbetsblad 3.2) undersökte du den korrekta tiden för att roboten skulle svänga sig 180°. Använd dig av denna tid och ställ in vänteikonen, där det nu står ett frågetecken (?).

Vad är den korrekta tiden för att få din robot att svänga exakt 180°?

Vad tycker du att ännu saknas från programmet och hur skulle du förbättra det?

www.meetedison.com

www.roboticswps.com.au

sida 47

# Lektion 7 Arbetsblad 7.4 - Att undvika hinder

Skriv följande program för att få Edison att kontinuerligt köra vidare efter att ha undvikit ett hinder.



Försök experimentera med olika tider (givna i sekunder) i den andra vänteikonen. Denna tid bestämmer hur länge roboten svänger (frågetecknet i bilden).

#### Vilken tror du är den bästa tiden för roboten att svänga på?

Varför är denna tidsinställning bäst? Vad skulle roboten kunna göra annorlunda för att ytterligare bli bättre?

# Lektion 7 Arbetsblad 7.5 – Detektera i höger och vänster led

Skriv följande program. Edison kommer nu kontinuerligt att köra vidare efter att ha detekterat hinder till höger eller vänster om sig.



I programmet använder vi en ny ikon, en "OM-slinga" (IF-loop). Det här är en väldigt viktig ikon eftersom den ger roboten möjlighet att ta beslut utan mänskligt inflytande. När en robot har denna egenskap kallas den autonom, eftersom den har artificiell intelligens.

En "om"-ikon frågar huruvida något är sant eller falskt. Om resultatet är sant tar programmet vägen utmärkt med ett rättecken. Om resultatet är falskt tar programmet vägen utmärkt med ett kryss.

Programmet har tre olika vägar det kan ta beroende på var ett hinder är. Förklara med egna ord vad dessa tre vägar får roboten att göra.

Inget hinder detekteras:

Hinder detekteras till höger:

Hinder detekteras till vänster:

#### Eftersom roboten kan ta beslut är den levande!? Vad tycker du?

www.meetedison.com

www.roboticswps.com.au

Nam	n	2
-----	---	---

# Lektion 8 Arbetsblad 8.1 – Sensor för linjeigenkänning

Edisonroboten är utrustad med en sensor som kan följa en linje. Sensorn består av två huvudsakliga elektroniska komponenter:

- 1. En röd Light Emitting Diode (LED)
- 2. Fototransistor (ljussensor)

LED-lampan lyser på underlaget som roboten kör på. Om du trycker två gånger på Edisons inspelningsknapp kan du se LED-lampan lysa. Undersök ljuset på underlaget genom att se hur det ändras när du lyfter roboten längre ifrån ytan. Jämför hur starkt ljuset på ytan är när ytan är svart eller vit.

Är ljuspölen starkare (reflekterar den mer ljus) på en svart eller vit yta? \_\_\_\_

Fototransistorkomponenten är en ljussensor och den mäter mängden ljus som ytan reflekterar.



Som du märkte i uppgiften i början så reflekteras det mer ljus från en vit yta än från en svart. Därför läser fototransistorn av mer ljus då Edison kör på ett vitt underlag än på ett svart. Detta gör det möjligt för roboten att programmeras att se skillnad på olika ytor och reagera på dessa. Ett svart underlag antas vara icke-reflekterande och ett vitt underlag antas vara reflekterande.

Hur tror du att linjesensorn skulle reagera på följande färgers ytor, som reflekterande eller icke-reflekterande? (Tips: ljuset är rött)

röd y	ta,	grön yta,	blå yta
-------	-----	-----------	---------

# Lektion 8 Arbetsblad 8.2 – Kör ända till en svart linje

Skriv följande program som får Edison att köra på ett vitt underlag (reflekterande) tills den korsar en svart linje (icke-reflekterande).



För att använda linjesensorn i ett program måste du först sätta på sensorn. Den första ikonen aktiverar alltså den röda LED-lampan.

Använd den svarta linjen på aktivitetsblad 8.1 eller rita en svart linje på ett papper. Du kan också använda dig av färgat tejp på ett bord. Kör roboten mot linjen och han kommer att stanna.

På aktivitetsblad 8.1 finns det också tre färgade linjer, en röd, en blå och en grön. Kör Edison mot de olika linjerna och se om han stannar.

Finns det en färg som Edison inte detekterar (ser) lika bra som de andra? Vilken färg är detta?

Varför tror du detta händer?

# Lektion 8 Arbetsblad 8.3 – Håll Edison innanför området



Skriv följande program som får Edison att hållas innanför ett avgränsat område.

Använd aktivitetsblad 8.2 som ett område eller skapa ditt eget med en tjock tusch på ett stort papper.

Du kan också använda svart tejp på ett vitt bord för att skapa ett inhägnat område.

Om du skapar ett stort område, se vad som händer om du lägger många robotar innanför området.

Experimentera också med olika hastigheter.

Hur snabbt kan Edison köra innan det uppstår problem?

Vad händer om Edison kör för fort?

Ν	а	m	۱r	ו:
---	---	---	----	----

# Lektion 8 Arbetsblad 8.4 – Följ en linje



Skriv följande program som får Edison att följa en svart linje.

Detta program använder en "OM"-ikon som ger roboten möjlighet att bestämma vad den gör. Om sensorn är på en reflekterande yta så skall roboten köra fram till höger med hastighet 4. Om sensorn är på en icke-reflekterande yta så kör roboten fram till vänster med hastighet 4. När roboten befinner sig på en vit yta kör den alltså bort från den och om den är på en svart yta gör den samma sak. Detta gör att roboten kör framåt och följer linjen.

Placera roboten innanför banan på aktivitetsblad 8.2 och låt roboten följa linjen.

Vilken väg kör roboten (med- eller moturs)?

Placera roboten utanför banan på aktivitetsblad 8.2 och låt roboten följa linjen.

Vilken väg kör roboten nu (med- eller moturs)?

Varför tror du att roboten kör åt olika håll runt banan?

# Lektion 8 Aktivitetsblad 8.1 – Sensor för linjeigenkänning

Använd detta aktivitetsblad för att testa ditt program från arbetsblad 8.2.

Du kan också kontrollera ditt svar på arbetsblad 8.1 med hjälp av linjerna nedan. För vilka färger stannar Edison?





### Lektion 8 Aktivitetsblad 8.2 - Linjesensor



# Lektion 9 Arbetsblad 9.1 – Att förstå variabler

En variabel är en liten del av datorns minne gjort för att lagra information. Vad som gör detta användbart är att informationen kan variera medan programmet kör, därifrån kommer namnet "variabel".

Variabler lagrar tal som 10, 106, 1482 o.s.v. och tillåter ett dataprogram att räkna, och detta är något som datorer är väldigt bra på.

Name	Range	Intial Value
Fred	0-255	162
Light_Level	+/- 32767	0
<new></new>		
4		

Edisonroboten har två olika sorters variabler som kallas "bytes" och "ord". Byte-variabler kan lagra tal från 0 till 255. Ord-variabler kan lagra större tal från -32767 till + 32767.

För att göra variabler lättare att använda namnger vi dem. Detta hjälper människor att minnas vilken sorts information som finns lagrad i dem. I EdWare kan du döpa dina variabler till nästan vad som helst. Du kan döpa en till "Per", men det kanske inte är så bra namn för att minnas vilken information som finns lagrad där. Ett bättre namn skulle kunna vara "Ljusniva". Denna typs namn hjälper dig att minnas vilken sorts information som du lagrar där.

Nu när du känner till variabler, låt oss skapa en i EdWare för nästa program.

Klicka på "Ny variabel" (Add Variable) i det övre vänstra hörnet och en liten box öppnas.

Edison EdWare	
File Program Edison He	elp
Zoom 100% 🔹 🗌	Add Variable Program Edison
control	Ádd Variable
-ḋ- flash	start

Add variable	<b>EX</b>
Variable name:	Light_Level
Variable range:	+/- 32767 🔹
Initial value (optional):	0
Delete	Cancel OK

Skriv in namnet "Ljusniva" på din variabel och välj sedan typen +/-32767 (ord). Sätt initialvärdet till noll. Klicka sedan på OK och din variabel kommer att läggas till i variabelfältet nere till höger.

#### Vilken typ av variabler skulle du använda för att lagra följande tal (byte eller ord)?

12 \_\_\_\_\_, 192 \_\_\_\_\_, 801 \_\_\_\_\_, -42 \_\_\_\_\_, 27,901 \_\_\_\_\_

N	ar	n	n	
---	----	---	---	--

### Lektion 9 Arbetsblad 9.2 – Ljusalarm

Skriv följande program som får Edison att spela upp ett alarm när lamporna i rummet tänds.



Ikonen för ljussensorn läser av ljusnivån med sin vänstra ljussensor och sätter det värdet i variabeln som heter Ljusniva.

Den första slingan använder matematik för att bestämma vad den skall göra.

Programmet upprepas tills värdet i variabeln Ljusniva är större än (>) 100.

När värdet i Ljusniva är större än 100 slutar slingan och programmet går till nästa loop som sätter på alarmet.

Placera Edison i ett mörkt rum och tryck på startknappen. När lamporna tänds börjar roboten larma.

Kan du tänka dig en situation där denna typs alarm skulle kunna vara användbart?

Vilka ändringar måste göras för att få programmet att bli ett "mörkhetsalarm"?

# Lektion 9 Arbetsblad 9.3 – Automatiska ljus



Skriv följande program som får Edison att slå på sina två LED-lampor när det blir mörkt.

Kör roboten genom en tunnel eller under en soffa och se hur lamporna slås på.

I detta program använder vi symbolen för "mindre än" (<) för att bestämma vad programmet gör. Om variabeln Ljusniva är "mindre än" 100 så tar programmet vägen med ett rättecken och slår på sina LED-lampor.

Experimentera med olika värden i OM-ikonen.

Vad händer om värdet är större än 100?

Vad händer om värdet är mindre än 100?

# Lektion 9 Arbetsblad 9.4 – Följa ljuset

Skriv följande program som får Edison att följa ljuset från en fackla/ficklampa.



Lys på Edison med en fackla/ficklampa och se att roboten kör mot ljuset.

Detta program utför beräkningar med två variabler. I "beräkna ord"-ikon subtraheras värdena på variablerna "Hoger\_ljus" och "Vanster\_ljus". Resultatet (svaret) sparas tillbaka till variabeln "Hoger\_ljus". Ta en titt på hur resultaten skulle kunna se ut:

	Hoger_ljus	Vanster_ljus	Beräkning	Resultat
Ljus till höger	200	100	200 - 100 =	+100
Ljus till vänster	100	200	100 - 200 =	-100

Från tabellen kan du se att när ljuskällan är till vänster är resultatet under noll (ett negativt tal). När ljuskällan är till höger är resultatet ett positivt tal.

Om-ikonen frågar: *Är resultatet mindre än noll?* Om detta är sant kör roboten till vänster (mot ljuset). Om det är falskt (resultatet > noll) kör roboten till höger, igen mot ljuset.

### Vad skulle hända om du bör ut mindre än-symbolen (<) till en större än-symbol (>)?

inamn:	Ν	а	m	n	i
--------	---	---	---	---	---

### Lektion I O Design arbetsblad I O. I – Mitt program

Filmen "Humans need not apply" gav insikt i hur robotar kan komma att användas i framtiden. Försök komma på en användbar uppgift roboten kan utföra och skriv sedan ett program som får roboten att utföra denna.

Här är några exempel:

- Räddningsrobot Roboten kör innanför ett område och söker en borttappad person (docka eller miniatyrmodell). När roboten lokaliserat personen ljuder ett alarm.
- Förarlös bil Roboten kör på en utmärkt väg (linje) utan att krocka med människor, andra bilar eller byggnader (leksaker).
- Stridsrobotar Skicka robotar i krig i stället för människliga soldater. Skapa en sumobrottningsring och programmera två robotar att leta efter varandra och sedan putta ut den andra ur ringen.

### I. Diskutera

Diskutera med din partner eller andra elever och försök komma på ett användbart sätt att utnyttja en robot med ett program ni kan programmera.

Vilka olika idéer kom ni på?

Beskriv varför en av idéerna inte gick att genomföra.

Namn	:			

### 2. Beskriv

Innan du börjar skriva ditt program skall du beskriva problemet som din robot skall lösa och hur den skall lösa detta (en lösning).

Problem:

### Lösning:

### 3. Skriv ditt program

Planera ditt program innan du börjar lägga till ikoner. Använd dig av de ikoner som du känner till, men kom ihåg att du också kan pröva på nya.

### 4. Misslyckades du?

Allting går inte alltid som det var meningen att det skulle gå. Ingen panik! Försök igen, men beskriv först vad som gick fel. Varför fungerade inte ditt program den första gången? *Försök igen, snart fungerar det!* 

Ν	а	m	۱r	1:
---	---	---	----	----

### Lektion IO Design arbetsblad IO.2 - Mitt program

### 5. Beskriv några av de programmeringsikoner som du använt

Rita och färglägg programmeringsikonerna nedan. Beskriv sedan vad de gör i ditt program.

Vad heter denna ikon?
Vad gör denna ikon?
Vad heter denna ikon?
Vad gör denna ikon?
Vad heter denna ikon?
Vad gor denna ikon?