



Ditt RobotikEdventyr

10 Lektionsplaner



De här lektionsplanerna är producerade i samarbete med RoboticsWPS



www.roboticswps.com.au

Innehåll

Möt Robotics WPS	4
Inledning.....	5
Lektionsöversikt.....	6
LEKTION 1: BEKANTA DIG MED EDISON	9
Lektion 1 Arbetsblad 1.1 – Möt Edison.....	16
Lektion 1 Arbetsblad 1.2 – Programmering med streckkoder.....	17
Lektion 1 Arbetsblad 1.3 – Möt EdWare.....	18
Lektion 1 Arbetsblad 1.4 – Ladda ner ett testprogram.....	19
LEKTION 2: ROBOTENS RÖRELSER – KÖR!.....	20
Lektion 2 Arbetsblad 2.1 – Kör roboten framåt.....	21
Lektion 2 Arbetsblad 2.2 – Kör roboten bakåt	22
Lektion 2 Arbetsblad 2.3 – Framåt och sedan bakåt	23
Lektion 2 Arbetsblad 2.4 – Hastigheter	24
Lektion 2 Aktivitetsblad 2.1.....	25
LEKTION 3: ROBOTENS RÖRELSER – SVÄNG!.....	26
Lektion 3 Arbetsblad 3.1 – Sväng 90° åt höger	27
Lektion 3 Arbetsblad 3.2 – Sväng 180° åt vänster.....	28
Lektion 3 Arbetsblad 3.3 – Sväng höger, sedan vänster	29
Lektion 3 Arbetsblad 3.4 – Minilabyrint.....	30
Lektion 3 Aktivitetsblad 3.1 – Sväng!.....	31
Lektion 3 Aktivitetsblad 3.2 – Minilabyrint.....	32
LEKTION 4: UTMANING OCH MEXIKANSKA VÅGEN	33
Lektion 4 Arbetsblad 4.1 – Utmaning	34
Lektion 4 Arbetsblad 4.2 – Mexikanska vågen	35
LEKTION 5: DESIGN 1 – MITT PROGRAM.....	36
Lektion 5 Design1Arbetsblad 5.1 – Mitt program.....	37
Lektion 5 Design 1 Arbetsblad 5.2 – Mitt program.....	38
LEKTION 6: ATT KÄNNA IGEN EN KLAPP.....	39
Lektion 6 Arbetsblad 6.1 – Blinka med LED som reaktion på en klapp	40
Lektion 6 Arbetsblad 6.2 – Kör som reaktion på en klapp	41

Lektion 6 Arbetsblad 6.3 – Dansa som reaktion på en klapp	42
LEKTION 7: DETEKTERA HINDER	43
Kalibrera sensorn för objektdetektion	44
Lektion 7 Arbetsblad 7.1 – Detektera hinder med den infraröda sensorn.....	45
Lektion 7 Arbetsblad 7.2 – Detektera hinder och stanna.....	46
Lektion 7 Arbetsblad 7.3 – Att undvika hinder	47
Lektion 7 Arbetsblad 7.4 – Att undvika hinder	48
Lektion 7 Arbetsblad 7.5 – Detektera i höger och vänster led	49
LEKTION 8: KÄNNA IGEN EN LINJE OCH FÖLJA DEN.....	50
Lektion 8 Arbetsblad 8.1 – Sensor för linjeigenkänning.....	51
Lektion 8 Arbetsblad 8.2 – Kör ända till en svart linje	52
Lektion 8 Arbetsblad 8.3 – Håll Edison innanför området.....	53
Lektion 8 Arbetsblad 8.4 – Följ en linje.....	54
Lektion 8 Aktivitetsblad 8.1 – Sensor för linjeigenkänning.....	55
Lektion 8 Aktivitetsblad 8.2 – Linjesensor.....	56
LEKTION 9: REAGERA PÅ LJUS	57
Lektion 9 Arbetsblad 9.1 – Att förstå variabler.....	58
Lektion 9 Arbetsblad 9.2 – Ljusalarm	59
Lektion 9 Arbetsblad 9.3 – Automatiska ljus.....	60
Lektion 9 Arbetsblad 9.4 – Följ ljuset.....	61
LEKTION 10: DESIGN 2 – MITT PROGRAM.....	62
Lektion10 Design arbetsblad 10.1 – Mitt program	63
Lektion 10 Design arbetsblad 10.2 – Mitt program	65
Tabell över elevernas framgång	66

Möt Robotics WPS

Michele Miller är direktör för och grundare av *Robotics WPS*, som startade sin verksamhet i 2009. Michele har arbetat som lärare både i New South Wales och i Queensland (Australien). Hon är lektor på University of Technology (UTS INSEARCH) i Sydney och hon har över 20 års lärarerfarenhet.

År 2010, då Michele undervisade i New South Wales, inbjöds hon att koordinera ett robotikprogram för UTS INSEARCH i Sydney. Ända sedan dess har teamet bakom *Robotics WPS* utvecklat en serie av kodningsworkshopar och erbjuder nu robotar och programmeringsworkshopar för skolor i Australien. Varje år får över 5000 elever ta del av *Robotics WPS* workshopar på ett eller annat sätt.

Robotics WPS erbjuder också konkret hjälp för skolor genom utbildningsdagar och fortbildningar.



Michele Miller från *Robotics WPS* och en ivrig robotikelev

Inledning

De här lektionerna är designade av Robotics WPS. Under de senaste 5 åren har de testats med många olika robotar i workshopar som Robotics WPS hållit. Nu har lektionerna testats med Edisonroboten och de var mycket omtänkta.

Du kan använda lektionsplanerna antingen som de är eller som inspiration för egna idéer. Lärare kan behöva göra ändringar för att möta elevernas kunskapsnivå på olika årskurser. Lektionssamlingen börjar med de lättare programmeringsikonerna, exempelvis grundläggande rörelser, och slutar med tidigare favoriter, exempelvis att följa en linje och sumobrottnig.

Lektionsplanerna inkluderar element av problemlösning och samarbetsförmåga och programmeringsprocessen är något elever på alla stadier kan njuta av.

Några elever kommer snabbt att greppa de grundläggande koncepten och snabbt vara redo för nästa aktivitet. Lektionsplanerna ger möjligheter till individualisering då eleverna själva kan övergå till nästa arbetsblad och arbeta självständigt. Om du är lite osäker rekommenderas att låta eleverna arbeta på i egen takt, med tiden ökar även ditt självförtroende och din tro på elevernas förmågor.

VO.2 Your EdVenture into Robotics / IO Lesson Plans

Lesson 2 Worksheet 1 - Drive the robot forward

Write the following program to drive the Edison robot forward. Use the activity sheet or marked lines on a desk or as start and stop markers.

Set the 'Seconds pass' (replace?) to an amount of time in seconds to have the robot start from behind the start line and stop before the finish line.

Minimum amount of time is 0.01 seconds.

Maximum amount of time is 327.67 seconds.

Try different amounts of time until your robot drives forward and stops just before the finish line.

What is the correct amount of time to get your robot from the start to the finish?

Describe what the robot is doing and why?

www.meetedison.com www.roboticswps.com.au page 20

Arbetsblad

Det finns ett antal arbetsblad tillgängliga för varje lektion. Arbetsbladen ger eleverna möjlighet att arbeta självständigt eftersom de innehåller instruktioner som eleverna kan följa.

Arbetsbladen innehåller också frågor för eleverna att svara på. Genom att svara på frågorna får eleverna demonstrera sitt lärande.

VO.5 Your EdVenture into Robotics / IO Lesson Plans

Lesson 3 Activity sheet 3.1 - Turning

Place the robot down as per the outline then play your turning program. Your robot should:

- Program 1 - Turn right from start 0° to end 90°
- Program 2 - Turn left from start 0° to end 180°
- Program 3 - Turn right from start 0° to end 90°, then turn left 270° to end 180°

www.meetedison.com www.roboticswps.com.au page 31

Aktivitetsblad

Några av lektionerna innehåller också aktivitetsblad. Aktivitetsbladet ger eleverna en plattform att snabbt testa sina program på.

Lektionsöversikt

De flesta lektionerna är designade för en 90 minuters lektion. Tidsåtgången varierar dock baserat på elevernas ålder och färdigheter. Alla lektionsplaner har en koppling till den australienska läroplanen. En lista på alla referenser finns på meetediton.com/.

Lektion 1: Bekanta dig med Edison

Teknologiska kunskaper – Eleverna bekantar sig med programmeringsomgivningen och hur man laddar ner ett program till Edison.

1. Bekanta dig med Edison med hjälp av streckkodsprogram
2. Öppna mjukvaruprogrammet och bekanta dig med hur ikonerna flyttas
3. Ladda ner ett testprogram

Lektion 2: Robotens rörelser – Kör!

Introduktion till sekventiell programmering – Eleverna lär sig hur roboten reagerar på kommandoikoner i sekvens och får greppa hur tid, hastighet och avstånd hör ihop.

- Program 1 – Kör roboten framåt (Arbetsblad 2.1)
- Program 2 – Kör roboten bakåt (Arbetsblad 2.2)
- Program 3 – Kör roboten framåt och bakåt (Arbetsblad 2.3)
- Program 4 – Hastigheter (Arbetsblad 2.4)

Lektion 3: Robotens rörelser – Sväng!

Sekventiell programmering och grundläggande geometri – Eleverna lär sig hur roboten reagerar på olika tidsinställningar och hur de kan kontrollera roboten medan den kör.

- Program 1 – Sväng höger (Arbetsblad 3.1)
- Program 2 – Sväng vänster (Arbetsblad 3.2)
- Program 3 – Sväng höger, sedan vänster (Arbetsblad 3.3)
- Program 4 – Minibana (Arbetsblad 3.4)

Lektion 4: En utmanande bana och den mexikanska vågen

Repetera lärdomar – Eleverna använder sig av kunskap från lektion 1 till 3 i två roliga öppna aktiviteter.

- Program 1 – En utmanande bana (Arbetsblad 4.1)
- Program 2 – "Mexikanska vågen robot style" (Arbetsblad 4.2)

Lektion 5: Design I – Mitt program

Kreativt tänkande och problemlösning – Eleverna skapar en egen utmaning och funderar på hur roboten skulle kunna ge dem en lösning.

1. Identifiera ett problem som roboten kan lösa eller gör en rolig rörelse till musik
2. Beskriv problemet eller rörelserna som roboten skall utföra
3. Skriv programmet och testa det
4. Misslyckande – Första försöket är aldrig lyckat, skriv ner och *Försök igen!*
5. Beskriv de programmeringsikoner som används och vad de gör
6. Demonstration – Eleverna demonstrerar sina program för klassen

Lektion 6: Att känna igen en klapp

Introduktion till input (sensorer) – Eleverna lär sig hur de får roboten att reagera på yttre stimuli (klappar). Lektionen innehåller också en rolig helklassaktivitet.

Program 1 – Blinka med LED som reaktion på en klapp (Arbetsblad 6.1)

Program 2 – Kör som reaktion på en klapp (Arbetsblad 6.2)

Program 3 – Dansa som reaktion på en klapp (Arbetsblad 6.3)

Lektion 7: Att detektera hinder

Introduktion till hur objekt kan detekteras och till artificiell intelligens – Eleverna programmerar roboten så att den gör val (artificiell intelligens) som reaktion på hinder i dess omgivning.

Att förstå infraröd detektion av hinder (Arbetsblad 7.1)

Program 1 – Detektera hinder och stanna (Arbetsblad 7.2)

Program 2 – Detektera hinder och undvik (Arbetsblad 7.3)

Program 3 – Detektera hinder och undvik i en slinga (Arbetsblad 7.4)

Program 4 – Detektera i höger och vänster led (Arbetsblad 7.5)

Lektion 8: Känna igen en linje och följa den

Robotbeteenden för industrin – Eleverna lär sig grundläggande robotbeteenden som liknar de som används i avancerade automatiserade fabriker och lagerbyggnader.

Sensorn för linjedetektion (Arbetsblad 8.1)

Program 1 – Kör ända till en svart linje (Arbetsblad 8.2)

Program 2 – Håll Edison innanför området (Arbetsblad 8.3)

Program 3 – Följa en linje (Arbetsblad 8.4)

Film – "Humans need not apply"

Lektion 9: Reagera på ljus

Mätningar av miljöfaktorer och programmera matematik – Eleverna lär sig hur man mäter ljusstyrka, sparar värdena i minnet och utför matematiska kalkyler för att kontrollera robotens beteenden.

Att förstå variabler (Arbetsblad 9.1)

Program 1 – Ljusalarm (Arbetsblad 9.2)

Program 2 – Automatiska ljus (Arbetsblad 9.3)

Program 3 – Följa ljuset (Arbetsblad 9.4)

Djuriska beteenden hos robotar

Lektion 10: Design 2 – Mitt program

Kreativt tänkande och problemlösning – Detta är den andra egna designen och eleverna får själv skapa en egen utmaning och beskriva hur roboten kan erbjuda en lösning.

1. Identifiera ett problem som roboten kan lösa
2. Beskriv problemet eller rörelser som roboten bör utföra
3. Skriv ett program och testa det
4. Misslyckande – Första försöket är aldrig lyckat, dokumentera och *Försök igen!*
5. Beskriv använda programmeringsikoner och vad de gör
6. Demonstration – Eleverna demonstrerar sina program för klassen

Kopplingar till NAPLAN tester (Australiens nationella elevtest)

Elever upplever att robotik är mycket motiverande. Motivation främjar en snabb inläring inom områden som naturvetenskaper, teknologi och matematik. Det finns också andra undervisningsämnen som kan dra nytta av en högre nivåns engagemang.

Under lektionerna ombeds eleverna skriva ner beskrivning av sina program och det som roboten gör. I arbetsbladen för de egna designerna får eleverna planera och skapa ett helt eget program. De skriver också ned och beskriver beteenden som de olika programmeringsikonerna skapar i deras program.

Robotics WPS erfarenhet har visat att många studerande har problem med att ta in information från visuell stimuli och omvandla detta till skriven text.

De här lektionsplanerna hjälper eleverna att överföra visuell stimuli från bilder, grafer och tabeller till skriven text. Detta är en viktig egenskap som behandlas i **NAPLANs test**.

Detta är bara ett av områdena var Robotics WPS har sett en tydlig förbättring av elevernas färdigheter. De har utvecklat en förmåga att förstå hur de skall överföra kunskapen de har tagit till sig kring sina program till skriven text.

Your EdVenture into Robotics / O Lesson Plans

Lesson 5 Design brief worksheet 5.2 - My program

5. Describe some of the programming icons in your program
Draw and colour in the programming icons below. Then describe what they do in your program.

	What is the name of this icon? <u>dual drive</u> What does this icon do? <u>In my program the dual drive icon drives my robot forward at speed 5.</u>
	What is the name of this icon? <u>event wait</u> What does this icon do? <u>In my program the event wait icon keeps my robot driving forward for 1.5 seconds. When this time has finished the program continues.</u>
	What is the name of this icon? <u>flash LED</u> What does this icon do? <u>In my program the flash LED icon turns on the left LED to draw attention to my dancing robot.</u>

www.meetedison.com
www.roboticswps.com.au

Ett exempel på ett ifyllt arbetsblad.

LEKTION 1 : BEKANTA DIG MED EDISON

Teknologiska kunskaper – Eleverna bekantar sig med programmeringsomgivningen och hur man laddar ner ett program till Edison.

Innan vi kan använda Edison behöver vi göra ett par saker.

1. Göra Edison redo för arbete
2. Lära känna Edison
3. Installera Edisons mjukvara "EdWare"
4. Kontrollera att allting fungerar genom att ladda ner och köra ett testprogram

Gör Edison redo



Kontrollera att batterierna ligger på rätt led.

Öppna batteriluckan och ta ut programmeringskabeln. Lägg in 4 st "AAA"-batterier. Kontrollera med hjälp av bilden att batterierna ligger rätt väg och lägg tillbaka luckan.

Sätt på Edison genom att föra brytaren till höger mot "på"-symbolen. Edisons röda LED-lampor kommer då att börja blinka.

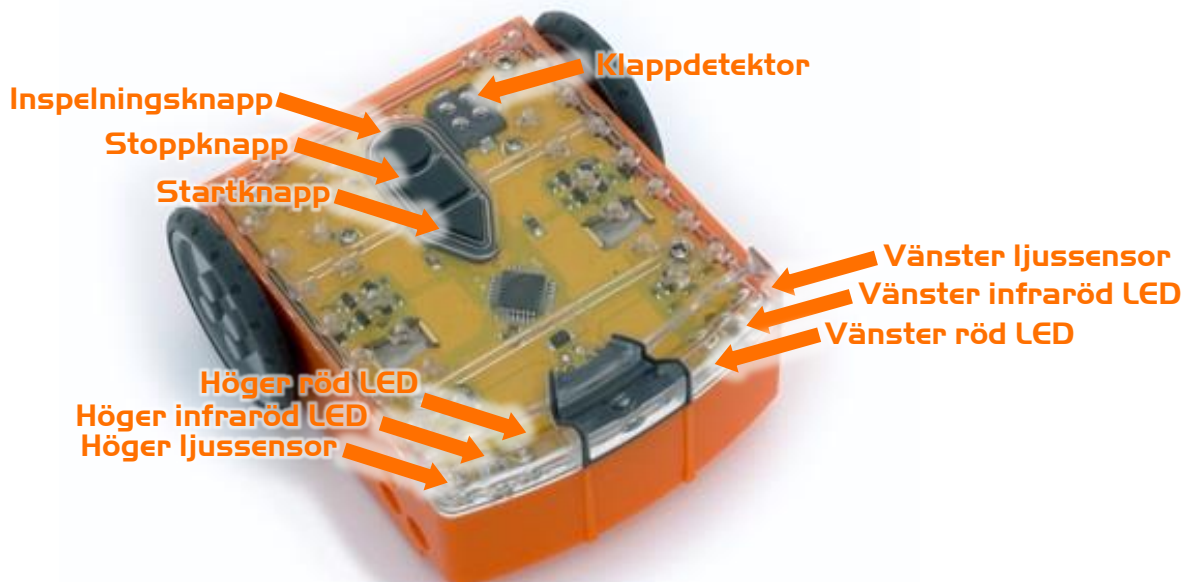
Edison är redo för arbete!



För knappen mot "på"-symbolen

Möt Edison (Arbetsblad 1.1)

Eleverna kommer att behöva vara bekanta med Edison och veta var alla sensorer ligger samt vad de tre knapparna gör. Be eleverna bekanta sig med roboten med hjälp av bilden på Arbetsblad 1. Det kan vara värt att ha detta arbetsblad tillgängligt för eleverna under alla 10 lektioner.



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 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.

Edisons strömknapp och sensor för linjedetektion.

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

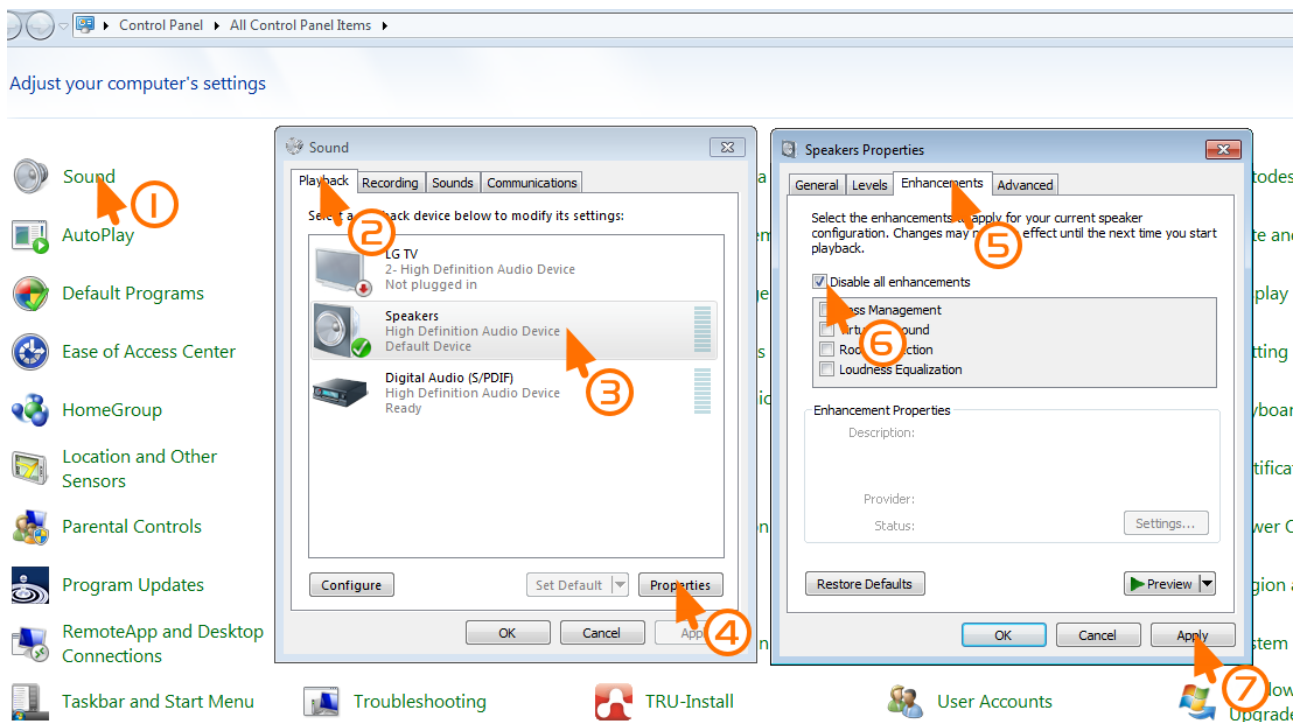


EdComm programmeringskabel

Installera EdWare

EdWare (Edisons programmeringsomgivning) finns tillgängligt för Windows, Mac, Linux, iOS, Android och Raspberry Pi. Du kan hitta installationsfiler och instruktioner för hur programmet installeras på webbsidan meetedison.com/downloads. Där finns filer för alla tillgängliga operativsystem.

I Windows bör ljudförstärkningen på datorn stängas av. Följ följande steg för att stänga av all förstärkning.



Hur man stänger av ljudförstärkning Windows 7 (Engelska)

Programmering med streckkoder (Arbetsblad 1.2)

För att inspirera eleverna och få dem intresserade av robotik och vad de kommer att lära sig, dela ut arbetsblad 2 tillsammans med robotarna. På arbetsbladet finns fyra streckkoder som Edison kan läsa av.

Låt studerande läsa av streckkoderna med Edison. Berätta för eleverna, för varje kod, vad Edison gör och vilka sensorer Edison använder sig av.

Kontrollera Edison med klappar

Placera Edison på en slät yta och tryck på startknappen (den triangulära). Klappa i händerna en gång nära Edison. Han kommer då att svänga höger. Klappa sedan i händerna två gånger och Edison kommer att köra framåt cirka 30 cm. Prova också på att knäppa till Edison med ditt finger, först en gång och sedan två gånger. Edison har en ljudsensor och använder sig av den för att reagera på klappar eller andra höga ljud.

Undvik hinder

Tryck på startknappen (triangeln) och se Edison närma sig hinder och vika åt sidan för att undvika en kollision. Edison använder sina infraröda lampor för att observera hinder.

Följa en linje

Placera Edison på ena sidan av din linje, (1,5 cm bred) så att sensorn är på ett vitt område. Tryck sedan på startknappen och se Edison följa linjen.

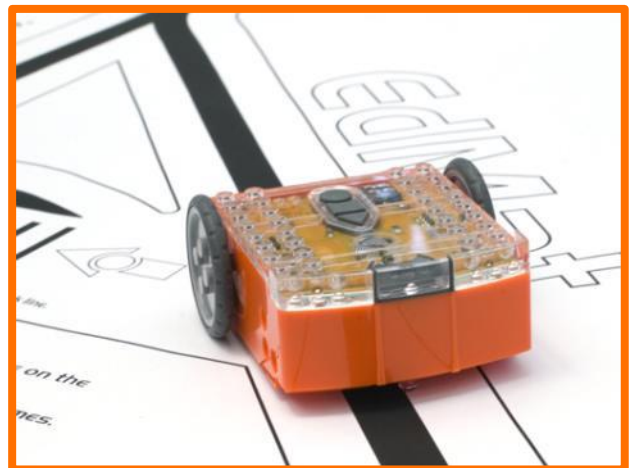
Edisons sensor lyser på ytan och mäter mängden ljus som reflekteras tillbaka. Vit färg reflekterar mycket ljus och svart väldigt lite, vilket ger Edison en möjlighet att "se" skillnad.

För att kunna följa linjen är Edison konstant missnöjd. När han är vid sidan av linjen vänder han höger för att hitta tillbaka till linjen, men om han är på linjen vänder han vänster för att komma bort från den.

Följa ett ljussken

Du behöver en ficklampa och en slät yta skyddad från starkt ljussken. Tryck på startknappen och rikta ljusskenet mot Edison. När han "ser" ljuskällan kommer han att köra mot denna. Genom att flytta ljuskällan kan du styrka vart Edison kör. *Påminner detta dig om något?*

Detta är ett av Edisons mer intressanta program. Det härmar beteende hos vissa flygande insekter. Du har kanske sett malar som svärmar kring ett ljussken en varm sommarkväll. Detta beteende kallas fototropism och finns t.ex. hos växter som vänder sig mot solen.



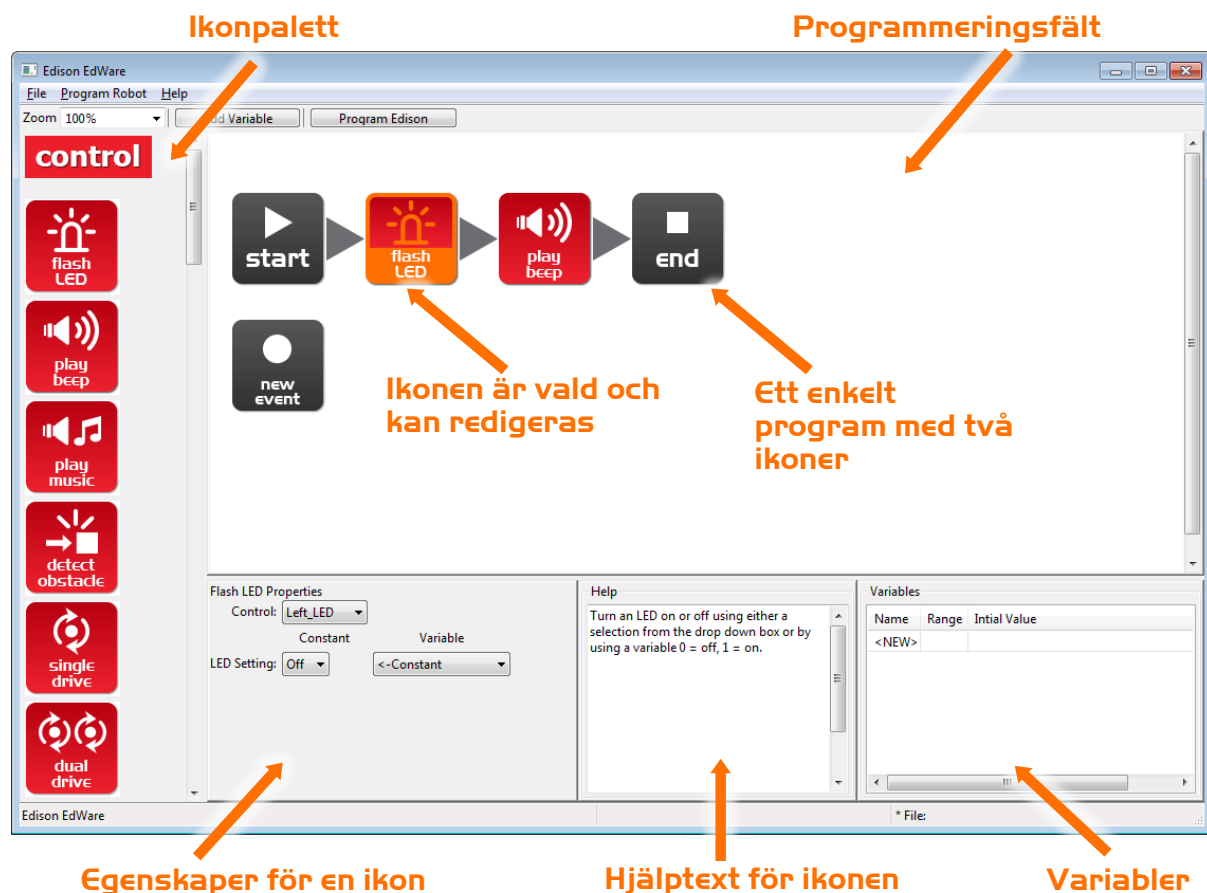
Edison line following

Möt EdWare (Arbetsblad 1.3)

Så fort eleverna är intresserade och motiverade av vad de sett Edison göra kan de börja använda EdWare.

För att bli bekanta med programmet, be eleverna öppna programmet och flytta ikoner från ikonpaletten till vänster till programfältet i mitten. Till att börja med behöver de inte laga fungerande program.

Så här ser EdWare för Windows ut. EdWare för de övriga operativsystemen påminner väldigt mycket om detta.



För att börja programmeringen, klicka på ikoner i paletten och dra ut dem på programmeringsfältet. Placera ikonerna mellan "start"- och "end"-ikonerna.

Välj en ikon. Inställningarna i egenskapsfältet bestämmer hur Edison skall reagera på just den ikonen. Experimentera lite med olika inställningar.

Använd hjälptexten som guide medan du programmerar. Allt du behöver veta om en ikon finns i detta fält.

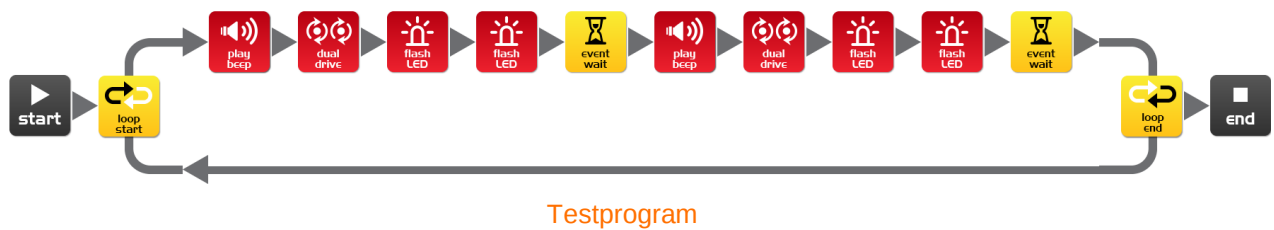
I fältet för variabler kan du skapa variabler och kontrollera små bitar av Edisons minne. Mer information om detta finns i lektion 9.

Testprogram (Arbetsblad 1.4)

Studerande kan nu öppna testprogrammet.

Filnamn: **Test Program.edw** (I Windows: C:\Program Files\EdWare\My Programs)

Testprogrammet ser ut så här:



Förklara för eleverna att ett program ser ut så här. Edison ser på en ikon i taget och gör vad ikonerna säger. Pilarna visar i vilken ordning Edison läser ikonerna. I programmet finns det en loop, så programmet upprepas gång på gång ända tills batteriet tar slut eller användaren avslutar programmet med stoppknappen.

Eleverna kan nu ladda ner testprogrammet till sina robotar.

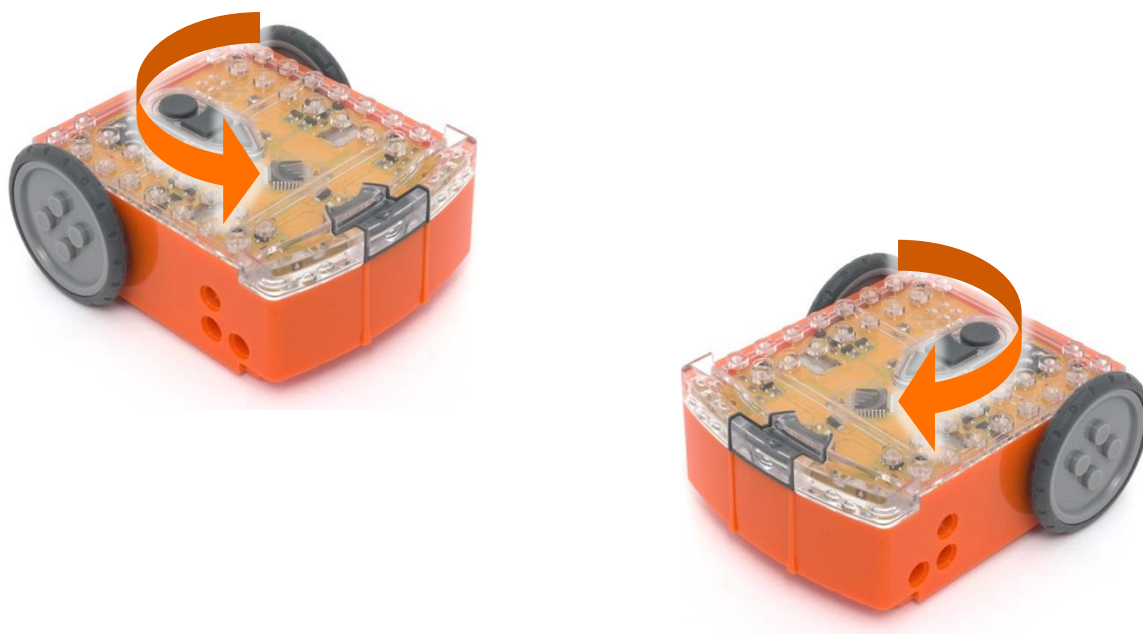
Koppla in EdComm-kabeln till jacket för hörlurar på datorn eller tabletten och **skruva upp volymen till full styrka**. Koppla andra änden av kabeln till Edison så som bilden visar.



För att ladda ner testprogrammet, utför följande steg:

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

Edison kommer nu att köra testprogrammet. Han snurrar både vänster och höger medan lamporna blinker och han piper.



Förklara för eleverna att roboten supersnabbt läser av ikonerna i kodsekvensen en åt gången i den ordning som pilarna bestämmer. Varje ikon berättar för roboten vad han skall göra. Förklara också att Edison inte kan läsa ikonerna i fel ordning eller hoppa över någon ikon.

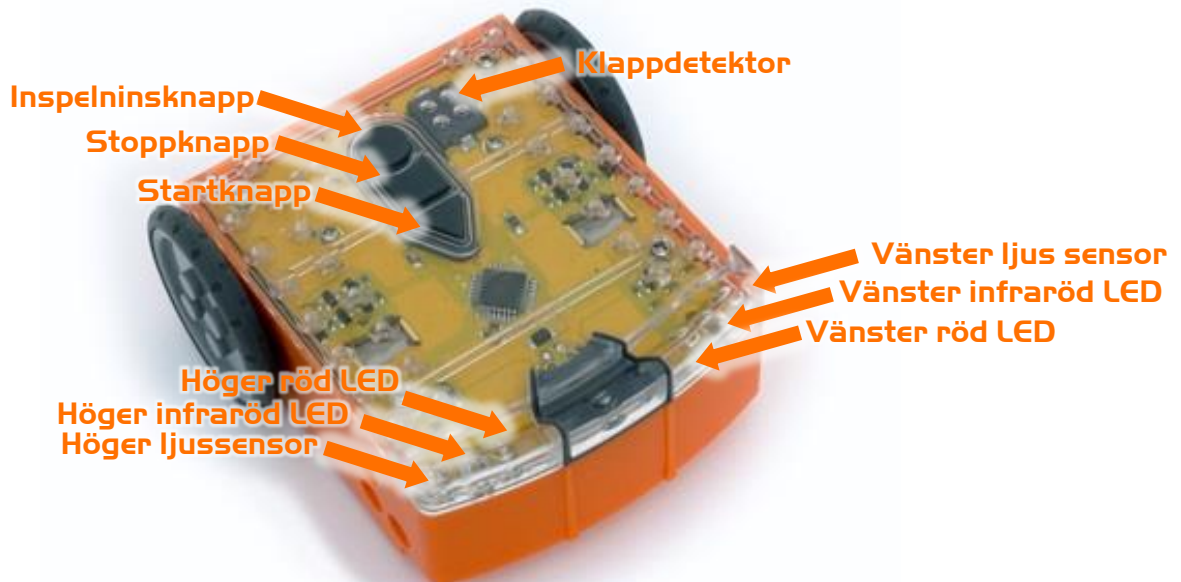
Förklara även åt eleverna att programmet färdas från datorn/tabletten till Edison via kabeln. Kabeln förvandlar ljudimpulsen från hörlursjacket till ljus. Edison tar sedan emot ljusimpulsen och sparar programmet i sitt minne.

Namn: _____

Lektion 1 Arbetsblad 1.1 – 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 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.

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.



EdComm programmeringskabel

Namn: _____

Lektion 1 Arbetsblad 1.2 – Programmering med streckkoder

Läs av streckkoden

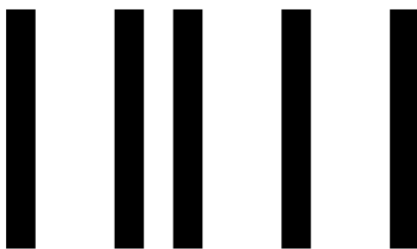
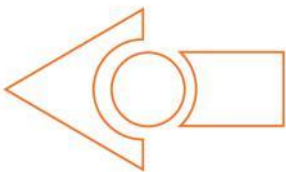
1. Placera Edison på höger sida om streckkoden så att han är riktad mot koden.
2. Tryck 3 gånger på inspelningsknappen
3. Edison kommer att köra framåt för att läsa av koden



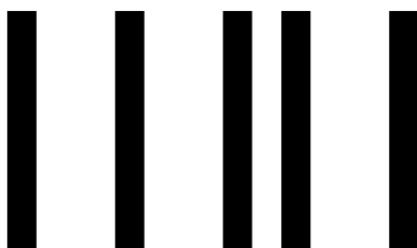
Streckkod – Klappkontrollerad körning



Streckkod- Undvik hinder



Streckkod – Följ linjen



Streckkod – Följ ljudkällan



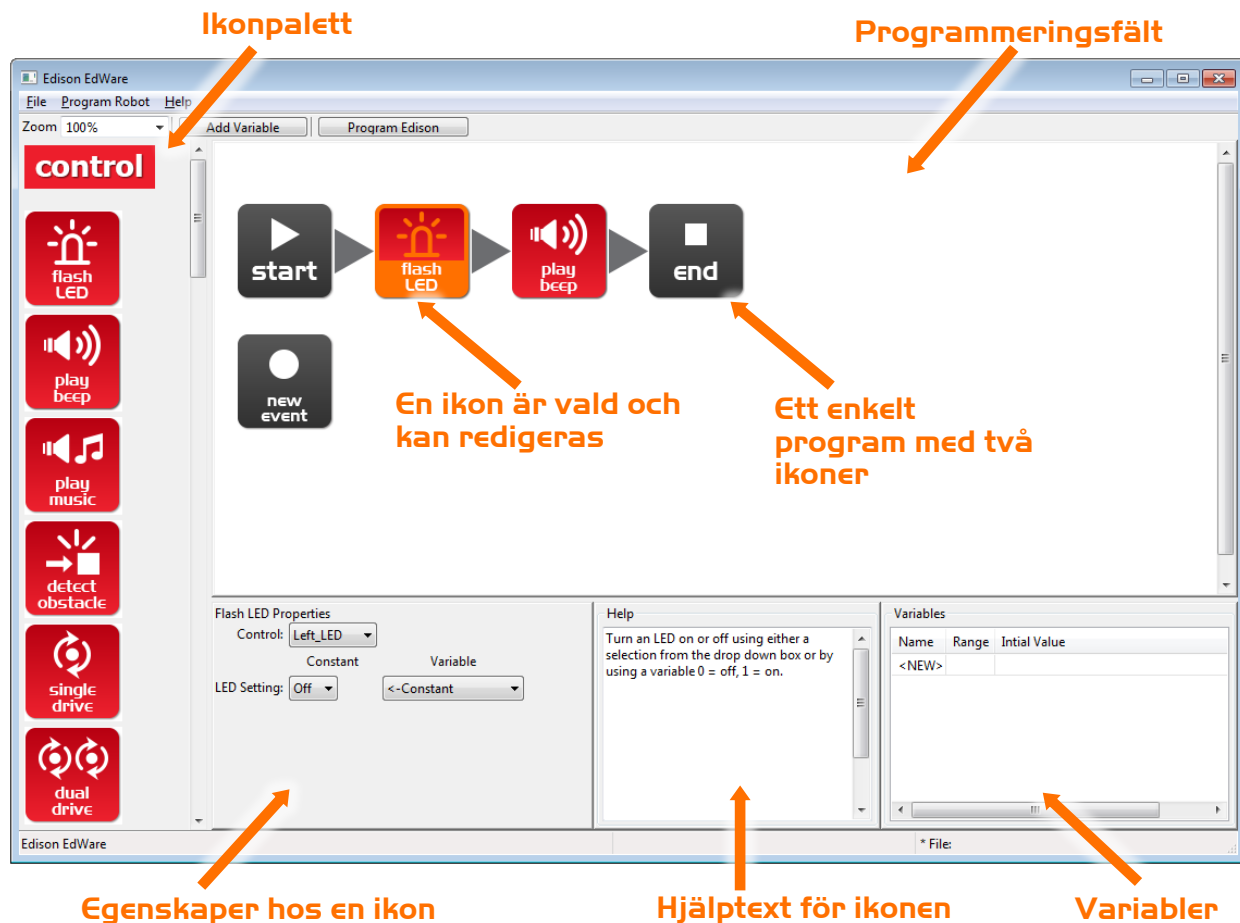
Beskriv vad roboten gör och varför den gör detta.

[illegible]

Namn: _____

Lektion 1 Arbetsblad 1.3 – Möt EdWare

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



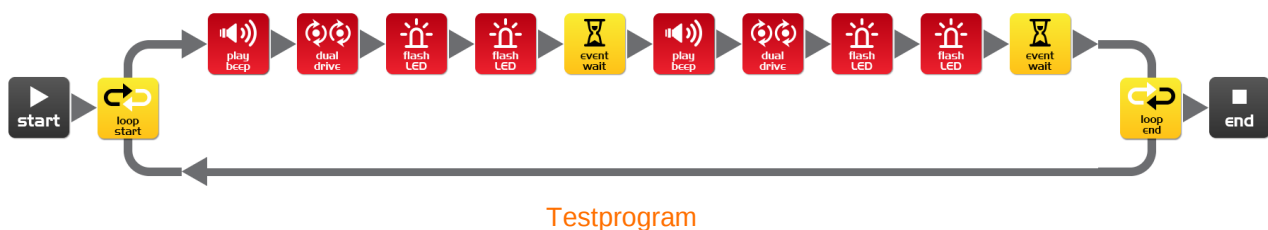
För att börja programmera skall du klicka på ikoner i paletten och dra ut dem på programmeringsfältet. Placera ikonerna mellan "start"- och "end"-ikonerna. Välj en ikon och ändra inställningarna i egenskapsfältet. Så bestämmer du hur Edison skall reagera på just den ikonen. Använd hjälptexten som guide 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?

Namn: _____

Lektion 1 Arbetsblad 1.4 – 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 ikonerna 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?

LEKTION 2: ROBOTENS RÖRELSER – KÖR!

Introduktion till sekventiell programmering – Eleverna lär sig hur roboten reagerar på kommandoikoner i sekvens och får undersöka hur tid, hastighet och avstånd hör ihop.

Under denna lektion använder eleverna basikonerna för att styra roboten och köra den framåt och bakåt.

Program 1 – Kör roboten framåt (Arbetsblad 2.1)

Be eleverna att skriva ett program som kör roboten framåt från ett startmärke till ett stoppmärke. Eleverna kan använda aktivitetsblad 2.1 eller färgat tejp som märken. Roboten måste stanna innan den kört över stoppmärket.

Program 2 – Kör roboten bakåt (Arbetsblad 2.2)

Be eleverna skriva ett program som kör roboten bakåt från ett startmärke till ett stoppmärke. Eleverna kan använda aktivitetsbladet som märken. Ett annat alternativ är färgat tejp på ett bord. Roboten måste stanna innan den kört över stoppmärket.

Program 3 – Kör roboten framåt och bakåt (Arbetsblad 2.3)

Be eleverna skriva ett program som kör roboten framåt från ett startmärke till ett stoppmärke och sedan kör bakåt tillbaka till startmärket. Som märken kan eleverna antingen använda arbetsbladet eller färgat tejp. Roboten måste stanna innan den kört över stoppmärket på vägen fram och innan den korsat startmärket på vägen tillbaka.

Program 4 – Hastigheter (Arbetsblad 2.4)

Be eleverna experimentera med tid, hastighet och avstånd.

Eleverna kan laga egna start- och stoppmärken.

Vad är den *kortaste* tiden på vilken roboten kan köra till stoppmärket och tillbaka?

Vad är den *längsta* tiden på vilket roboten kan köra till stoppmärket och tillbaka?

Eleverna kan lägga till blinkande LED-lampor och ljus till sina program om de vill vara kreativa.

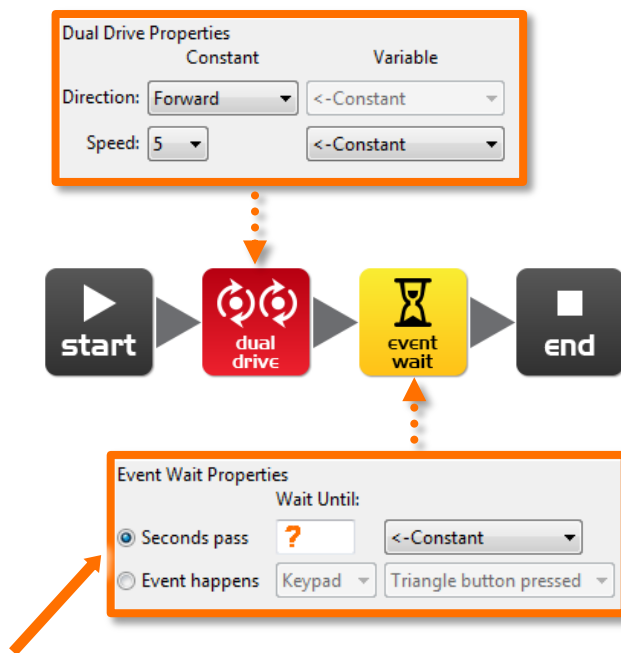
Varje arbetsblad ber eleverna att beskriva vad roboten gör samt varför. Ett typiskt svar kan se ut som följande:

Roboten kör framåt eftersom programmet har en ikon som säger att den skall köra framåt. Den kör till mållinjen och stannar eftersom tiden i vänteikonen är ställd på 0,35 s och det är så här länge det tar för roboten att köra så långt.

Namn: _____

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.



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?

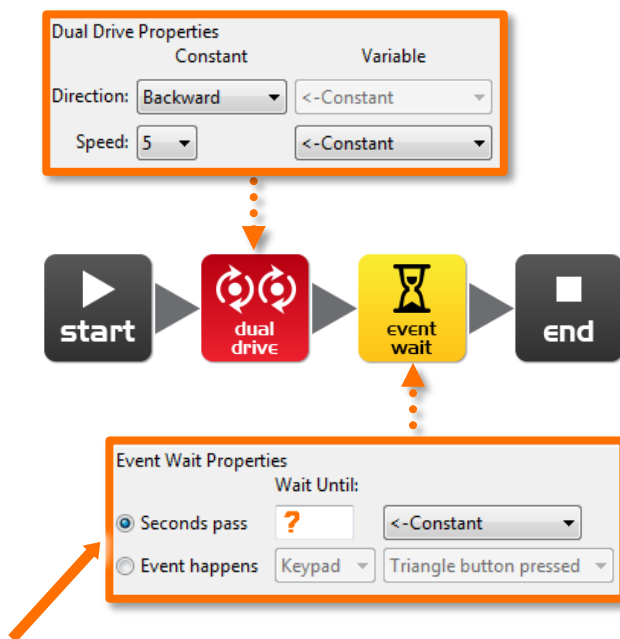
Beskriv vad roboten gör och varför.

www.meetedison.com

Namn: _____

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.



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.

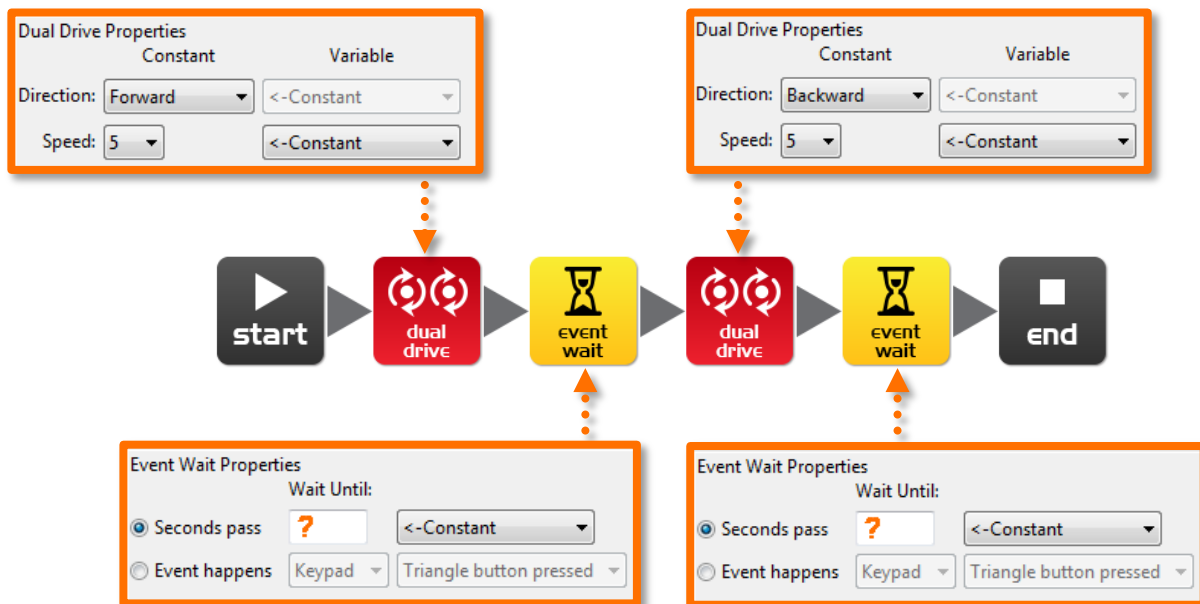
Vad är den korrekta tiden som får din robot att köra från startpunkten till mållinjen?

Beskriv vad roboten gör och varför.

Namn: _____

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?

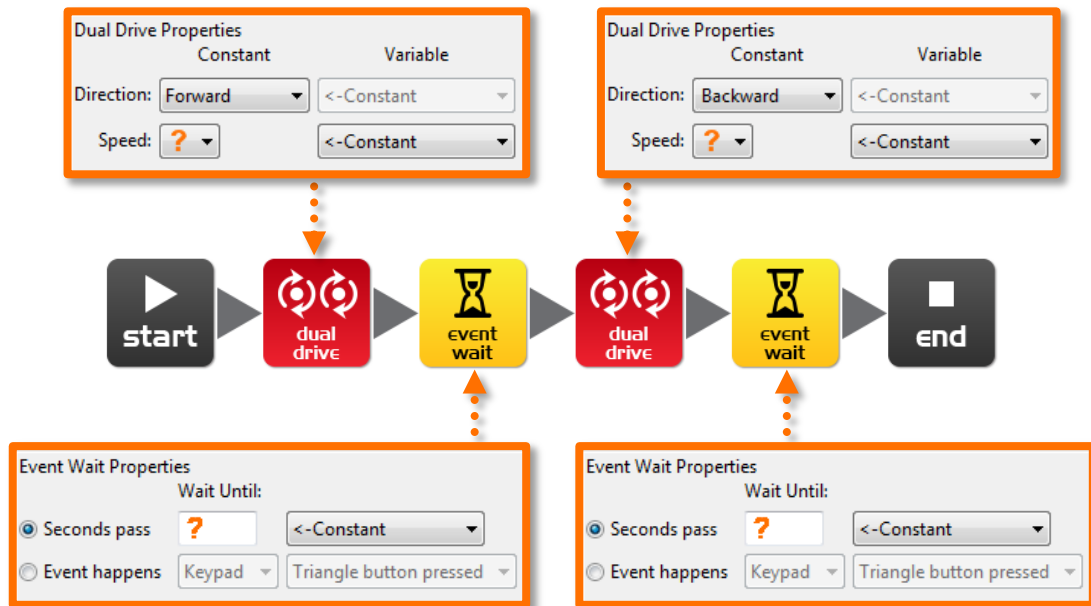
Framåt _____ Bakåt _____

Beskriv vad roboten gör och varför.

Namn: _____

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.



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 LED-lamporna, 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 _____

Namn: _____

Lektion 2 Aktivitetsblad 2.1



MÅLLINJE



STARTLINJE

LEKTION 3: ROBOTENS RÖRELSER – SVÄNG!

Sekventiell programmering och grundläggande geometri – Eleverna lär sig hur robotens reagerar på tid och geometri och hur de kan kontrollera roboten medan den kör.

I denna lektion använder sig eleverna av de grundläggande för att svänga roboten olika vinklar (90°, 180° och 270°). Sedan använder de kunskap från lektion 2 för att skapa en sekvens av rörelser.

Program 1 – Sväng höger (Arbetsblad 3.1)

Be eleverna skriva ett program som svänger roboten 90° åt höger. Använd antingen aktivitetsblad 3.1 eller färgat tejp på bordet som märken.

Program 2 – Sväng vänster (Arbetsblad 3.2)

Be eleverna skriva ett program som svänger roboten 180° åt vänster. Använd antingen aktivitetsblad 3.1 eller färgat tejp på bordet som märken.

Program 3 – Sväng höger, sedan vänster (Arbetsblad 3.3)

Be eleverna skriva ett program som svänger roboten först 90° åt höger och sedan 270° (90° + 180°) åt vänster. Använd antingen aktivitetsblad 3.1 eller färgat tejp på bordet som märken.

Program 4 – Minibana (Arbetsblad 3.4)

Be eleverna skriva ett program som kan navigera sig igenom en liten bana. Använd aktivitetsblad 3.2 eller skapa en egen bana med färgat tejp på bordet. Eleverna kommer att använda kunskap från lektion 2 och 3 för att slutföra denna utmaning.

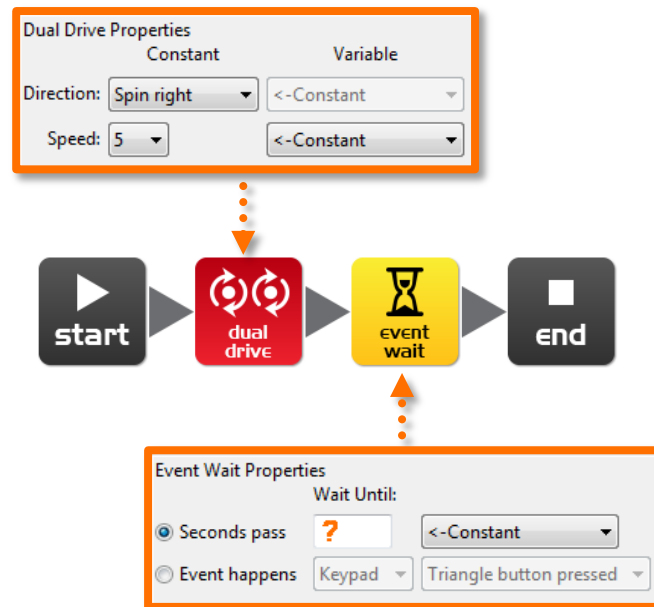
Varje arbetsblad ber eleverna att beskriva vad roboten gör och varför. Ett typiskt svar kan se ut som följande:

Roboten svänger eftersom programmet har en körikon som ber den svänga vänster. Den svänger 90° eftersom den har en vänteikon med tiden 0,35 sekunder och det tar så länge för roboten att svänga 90°.

Namn: _____

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.



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.

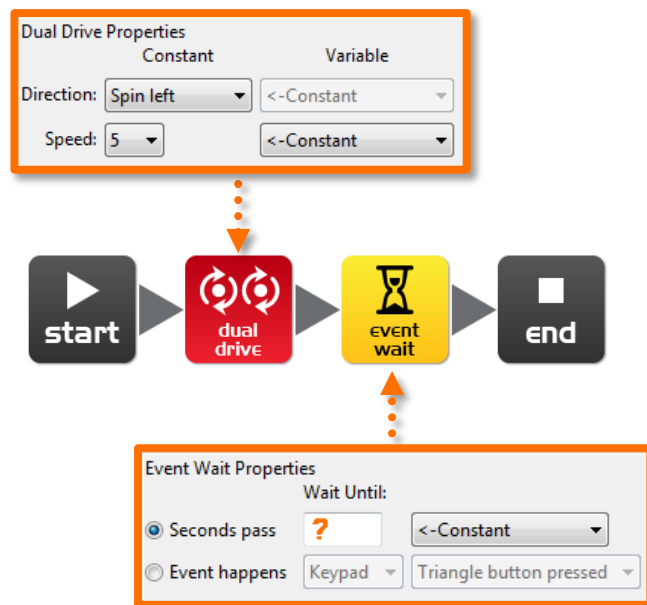
Vad är den korrekta tiden för att få din robot att svänga sig 90°?

Beskriv vad roboten gör och varför.

Namn: _____

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.



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.

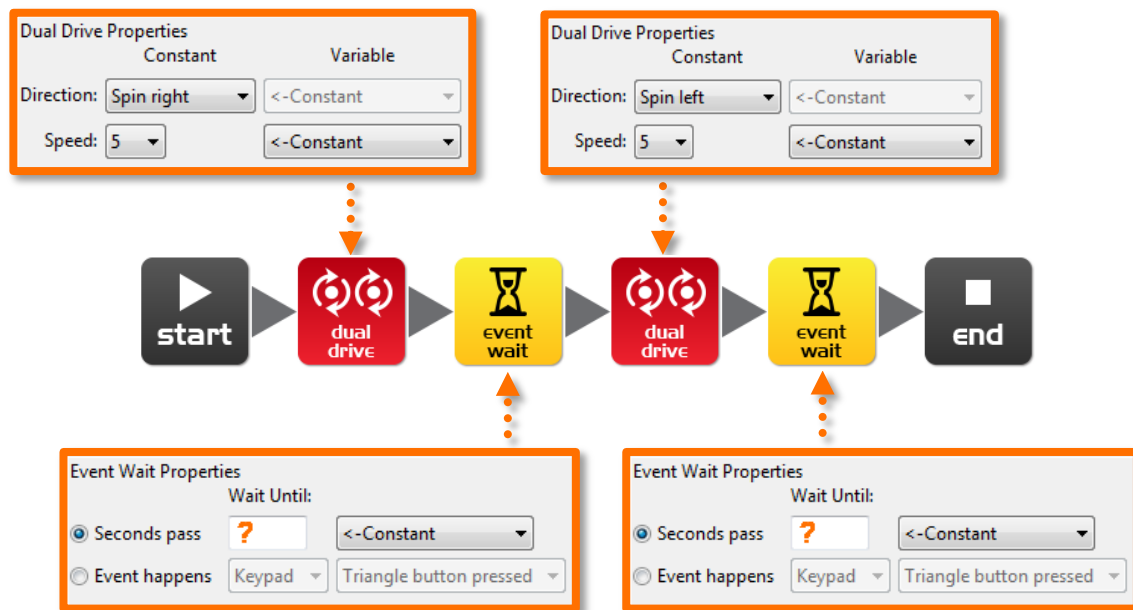
Vad är den korrekta tiden för att få din robot att svänga sig 180°?

Beskriv vad roboten gör och varför.

Namn: _____

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 når 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?

Namn: _____

Lektion 3 Arbetsblad 3.4 – Minilabyrint

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: _____

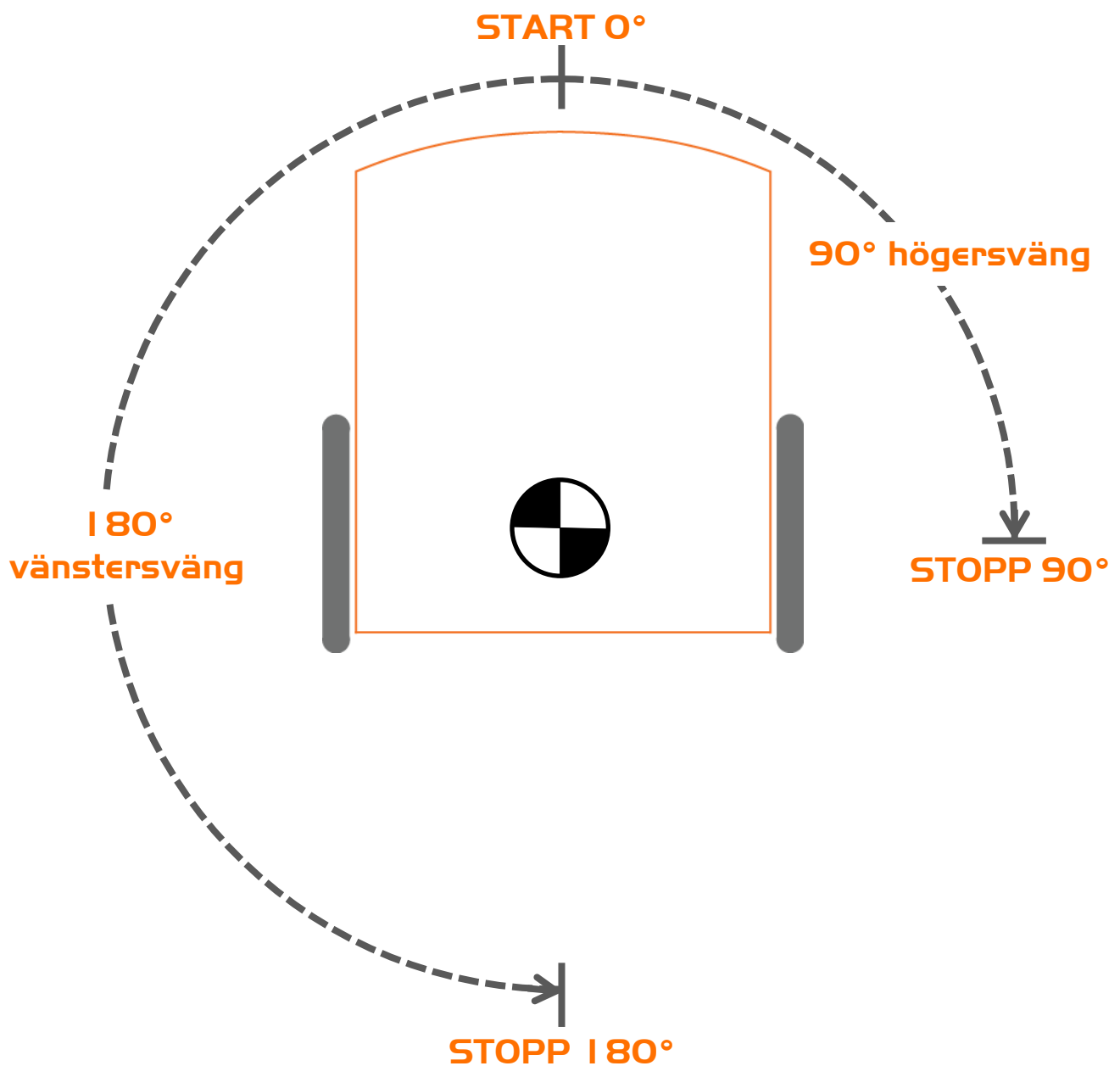
Lektion 3 Aktivitetsblad 3.1 – Sväng!

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°



Namn: _____

Lektion 3 Aktivitetsblad 3.2 – Minilabyrint



MÅLLINJE

STARTLINJE

LEKTION 4: UTMANING OCH MEXIKANSKA VÅGEN

Repetera lärdomar – Eleverna använder sig av kunskap från lektion 1 till 3 för att utföra två roliga öppna aktiviteter.

Under denna lektion utför eleverna två programmeringsaktiviteter.

Program 1 – Körutmaning

Eleverna väljer en utmaning för sin robot och skriver sedan ett program för den. Några exempel är:

- Kör runt ett hinder, ex. en kopp eller en penal
- Kör runt kanten på ett bord utan att köra över kanten
- Skapa en labyrinth på ett stort papper som roboten kan köra igenom
- Skapa en labyrinth med en byggsats och be roboten köra igenom den

Program 2 – “Mexikanska vågen robot style”

Den mexikanska vågen är en rolig aktivitet för hela klassen. Allas robotar kör samma program men de börjar inte samtidigt. Resultatet liknar en mexikansk våg eller koreografisk dans.

Eleverna skriver ett kort program med en “cool” sekvens av robotrörelser. Uppmuntra dem även att använda sig av ljus och ljud i sina program.

Klassen (eller läraren) väljer det program som är ”bäst”. Sedan delar ni det programmet med alla övriga elever. Varje elev lägger till en vänteikon i början av programmet.

För att skapa effekten av en mexikansk våg programmerar varje elev in olika tider i den nya vänteikonen. Exempelvis:

Robot 1 – Väntar 0 sekunder
Robot 2 – Väntar 0,2 sekunder
Robot 3 – Väntar 0,4 sekunder
Robot 4 – Väntar 0,6 sekunder
Robot 5 – Väntar 0,8 sekunder
Robot 6 – Väntar 1,0 sekunder
Robot 7 – Väntar 1,2 sekunder
Robot 8 – Väntar 1,4 sekunder
Robot 9 – Väntar 1,6 sekunder
Robot 10 – Väntar 1,8 sekunder

Robot 1 till 10 placeras på en rad i samma ordning som deras tidsfördröjningar. Alla trycker på startknappen samtidigt och betraktar sedan uppvisningen.

OBS: Apple tillåter inte import av filer till iOS, så denna aktivitet är kanske inte möjlig om ni använder er av iPads.

Namn: _____

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?

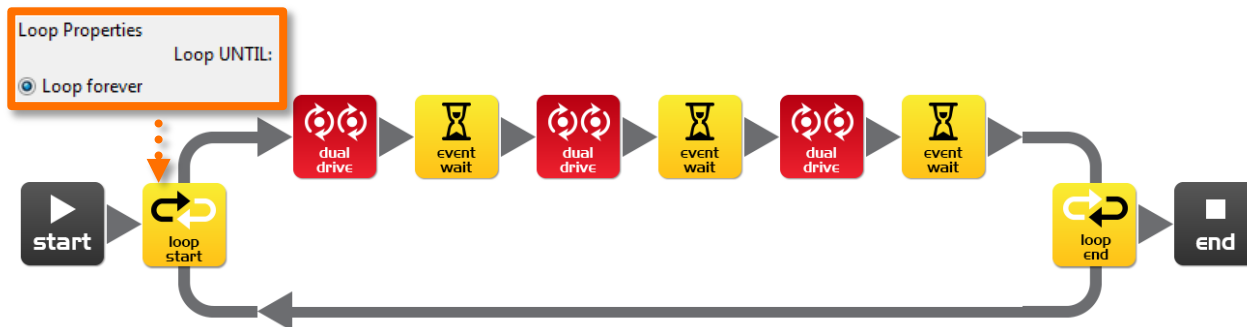
Namn: _____

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.

LEKTION 5: DESIGN I – MITT PROGRAM

Kreativt tänkande och problemlösning – Eleverna skapar en egen utmaning och funderar på hur roboten skulle kunna ge dem en lösning. Denna lektion kan ta två lektioners tid, alternativt kan planeringsskedet vara **hemläxa**. Eleverna får välja en egen rubrik för sitt arbete, beskriva syftet med programmet och förklara var i det verkliga livet programmet skulle kunna användas.

1. Identifiera ett problem som roboten kan lösa eller skapa en rolig dans

I framtiden kommer robotar att vara våra hjälpredor. Vi har redan robotdammsugare, så framtiden ligger inte långt härifrån. Vilka hjälpsamma eller underhållande uppgifter kan elevernas robotar utföra?

Till exempel:

- Dansa till musik – roboten dansar till elevernas favoritmusik
- Dammsugare – eleverna kan rita en planritning för ett vardagsrum på ett stort papper och programmera roboten så att den "dammsuger" hela rummet.
- Vaktrobot – eleverna kan definiera ett säkert område (ett område på ett papper eller ett litet objekt). Robotens jobb är att vakta området genom att köra runt objektet.

2. Beskriv problemet eller rörelserna som roboten måste utföra

Med stöd av designarbetsbladet **beskriver eleverna problemet** som just deras robot skall lösa och **hur** robotens program **löser** problemet.

Exempel: **Problemet är...** "Människor kan dansa bra, men deras rörelser är långsamma och det kan vara svårt att dansa i takt." **Min robot löser detta genom att...** "De gör snabba rörelser och hålls i samma takt som de andra robotarna som dansar samma dansrutin. Detta kommer att vara intressantare att titta på än människor som dansar."

3. Skriv ett program och testa det

Eleverna skriver sina program och testar dem.

4. Misslyckande

Det första försöket kommer inte att vara helt lyckat. Några elever kan behöva extra uppmuntran att *försöka igen!* Misslyckande är en normal del av programmering (och alla andra ingenjörssaktiviteter). Thomas Edison är berömd för att ha misslyckats 10 000 gånger innan han lyckades med uppfinnandet av glödlampan. Eleverna kan beskriva sina hinder på designarbetsbladet.

5. Beskriv de använda programmeringsikonerna och vad de gör

Eleverna väljer några ikoner de använt och beskriver dessa (och deras uppgift) på arbetsblad 5.2. Printa fler arbetsblad om studerande skall beskriva flera ikoner.

6. Demonstration

Eleverna demonstrerar sina program och berättar om sin idé, programmet, problem de stött på och hur de löst problemen.

Namn: _____

Lektion 5 Design I Arbetsblad 5. I – 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

1. 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?

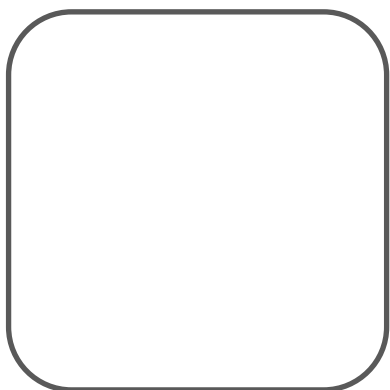
Allting går inte som man tänkt sig den första gången. Ingen panik! Försök igen, men beskriv först vad som gick fel. **Varför fungerade inte ditt program den första gången?** Kom ihåg att Thomas Edison misslyckades 10 000 gånger innan han lyckades med att uppfinna sin berömda glödlampa. Så, *försök igen!*

Namn: _____

Lektion 5 Design I Arbetsblad 5.2 – Mitt program

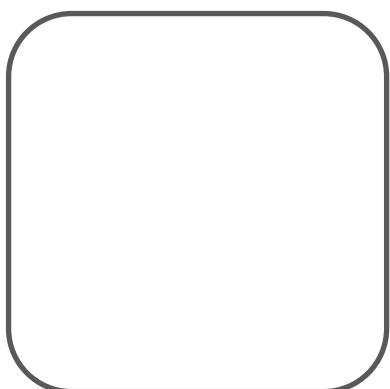
5. Beskriv några av de ikoner som används i ditt program

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



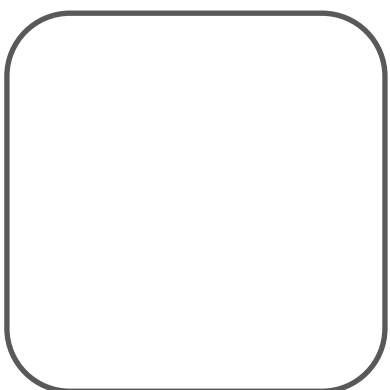
Vad heter denna ikon? _____

Vad gör denna ikon? _____



Vad heter denna ikon? _____

Vad gör denna ikon? _____



Vad heter denna ikon? _____

Vad gör denna ikon? _____

LEKTION 6: ATT KÄNNA IGEN EN KLAPP

Introduktion till input (sensorer) – Eleverna lär sig hur de får roboten att reagera på yttre stimuli (klappar).

Under denna lektion lär sig studerande hur en robot kan känna av yttre stimuli. Dessutom behandlas hur man programmerar robotens respons och hur man skapar ett beteende för roboten.

Program 1 – Blinka med LED som respons på en klapp (Arbetsblad 6.1)

Eleverna bekantar sig med att programmera roboten att vänta på att en händelse sker innan programmet fortsätter. I detta fall är händelsen ett starkt ljud som en klapp.

Program 2 – Kör som reaktion på en klapp (Arbetsblad 6.2)

Eleverna använder sig nu av kunskap från tidigare lektioner och får roboten att köra framåt som reaktion på en händelse (en klapp).

Program 3 – Dansa som reaktion på en klapp (Arbetsblad 6.3)

Eleverna skapar egna rörelser som reaktion på klappar. De har två olika möjligheter:

1. Skapa en robotdans med många olika dansmanövrar.
2. Skapa ett program som kör roboten genom en hinderbana som reaktion på klappar.

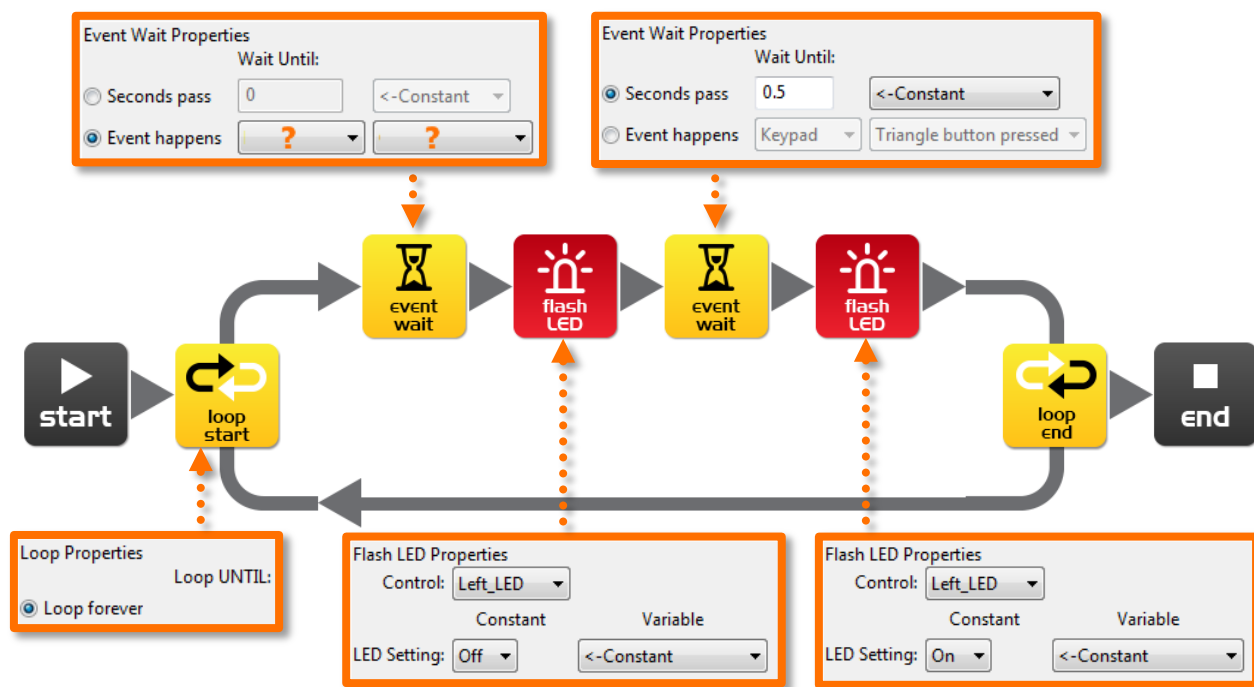
Rolig klassaktivitet

Välj ut den bästa dansrutinen och låt alla andra elever programmera sina robotar med den. Placera alla robotar på samma ställe och låt eleverna klappa i takt för att få robotarna att dansa tillsammans. Du kommer att behöva räkna in dem i klappandet genom 1, 2, 3, 4... Klapparna startar på 5, och sedan fortsätter de på 6, 7, 8...

Namn: _____

Lektion 6 Arbetsblad 6.1 – 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.

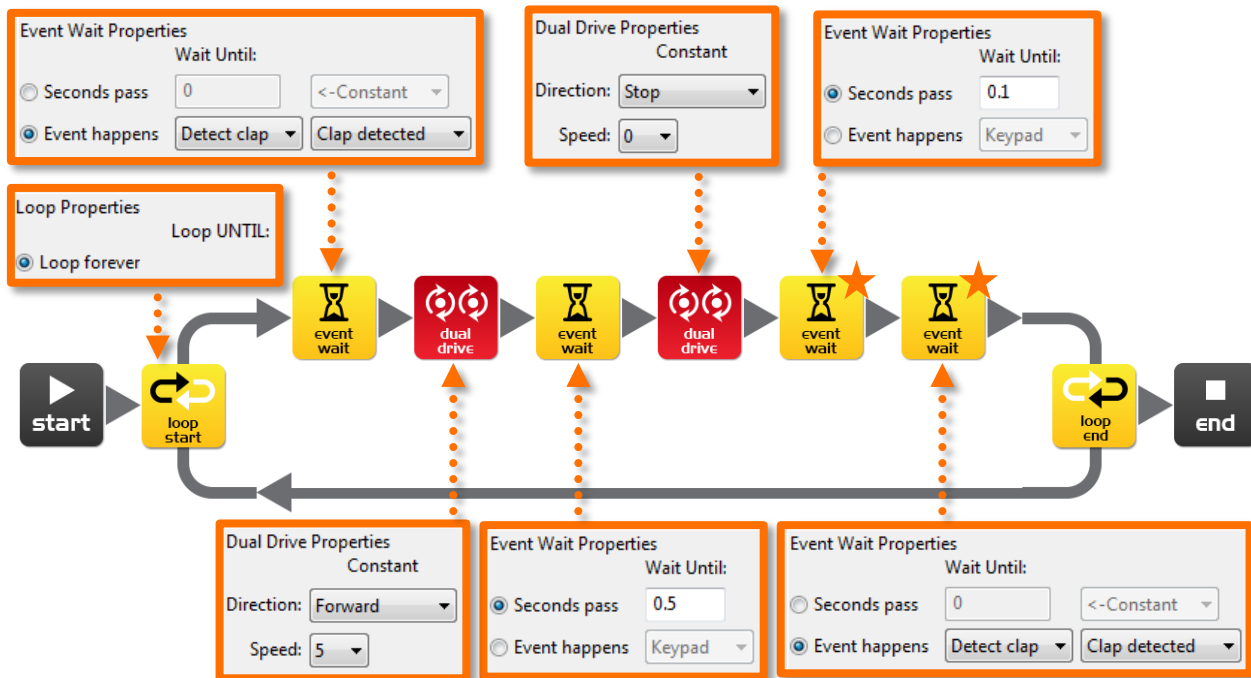
På vilket avstånd detekterar roboten din klapp?

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

Namn: _____

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 köriken 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 _____

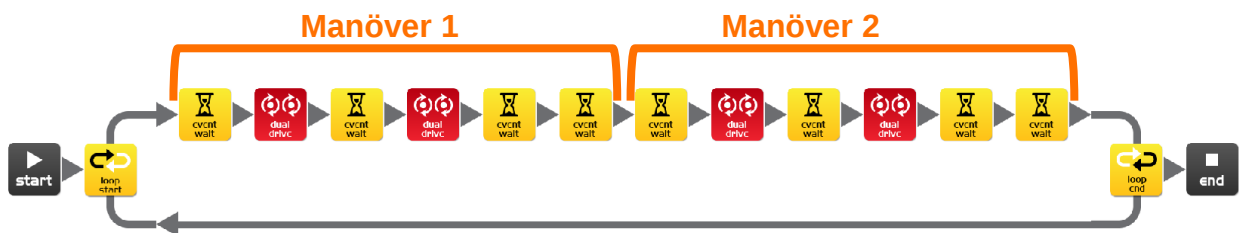
Experiment 2: riktning _____, tid _____

Experiment 3: riktning _____, tid _____

Namn: _____

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.

LEKTION 7: DETEKTERA HINDER

Introduktion till hur objekt kan detekteras och till artificiell intelligens – Eleverna programmerar roboten så att den gör val (artificiell intelligens) när den detekterar hinder i sin omgivning.

Att förstå infraröd detektion av hinder (Arbetsblad 7.1)

Eleverna bekantar sig med robotens teknologi för att detektera hinder. Edison använder infrarött ljus för att märka hinder i sin väg.

Program 1 – Detektera hinder och stanna (Arbetsblad 7.2)

Eleverna skriver ett program som kör roboten tills den märker ett hinder som står i vägen. Robotens stannar då för att undvika en kollision.

Program 2 – Detektera hinder och undvik (Arbetsblad 7.3)

Eleverna skriver ett program som kör roboten tills den märker ett hinder som står i vägen. Roboten svänger då för att undvika en kollision.

Program 3 – Detektera hinder och undvik i en slinga (Arbetsblad 7.4)

Eleverna skriver ett program innehållande en kontinuerlig slinga. Roboten kör då ända tills den märker att det står ett hinder i vägen. Roboten svänger för att undvika kollision. Edison fortsätter sedan köra, tack vare slingan, och söker andra hinder att undvika.

Program 4 – Detektera i höger och vänster led (Arbetsblad 7.5)

Eleverna skriver ett program innehållande en kontinuerlig slinga. Programmet reagerar på huruvida det finns ett hinder till höger eller vänster om Edison. För att åstadkomma detta används en "OM"-slinga (IF-loop).

En OM-slinga är viktig då den ger roboten möjlighet att ta beslut utan mänskligt inflytande. När detta sker kallas roboten för en **autonom robot**, eftersom den nu har artificiell intelligens. Det här är ett väldigt mångfacetterat koncept, är roboten nu levande?

Denna lektion kan också användas för att stimulera en diskussion kring förarlösa robotbilar. Bilarna använder sensorer för att undvika kollisioner med människor, byggnader och andra bilar.

Om sensorerna för objekt-detektion är för känsliga eller inte känsliga nog kan eleverna rekalkibrera dem med hjälp av instruktionerna på följande sida.

Namn: _____

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 streckkod 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.

www.meetedison.com

Namn: _____

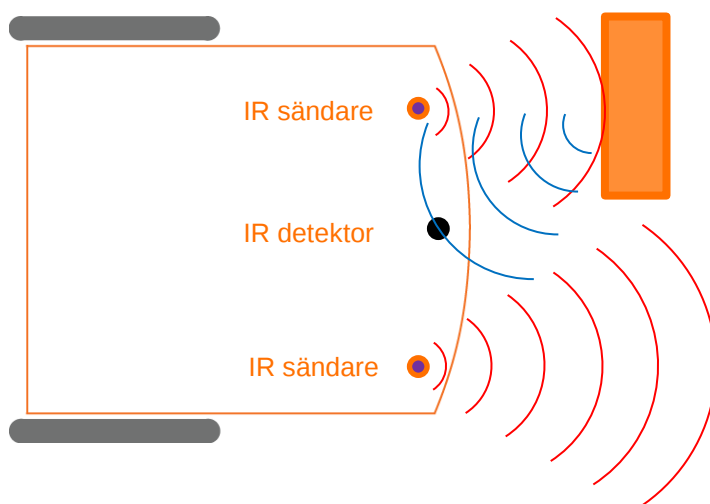
Lektion 7 Arbetsblad 7.1 – Detektera hinder med den infraröda sensorn

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.

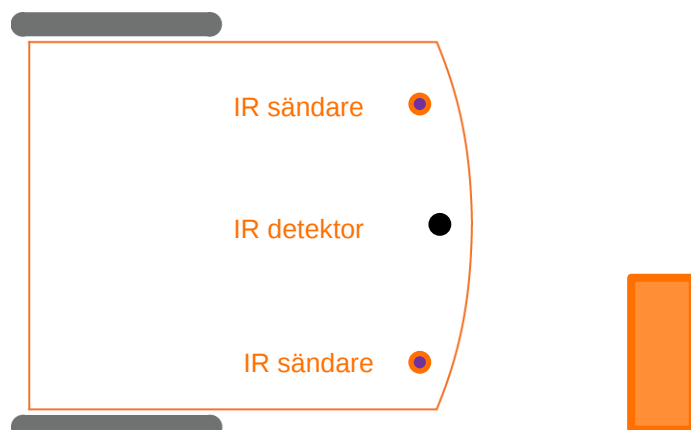
För att Edison skall kunna märka hinder skickar den ut infrarött ljus framför sig. Edison har lampor både på den vänstra och högra sidan. Om ljuset reflekteras från ett hinder, ex. en vägg, så detekteras det reflekterade ljuset av Edisons infraröda detektor. Detektorn ligger längst fram i mitten på roboten.

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.



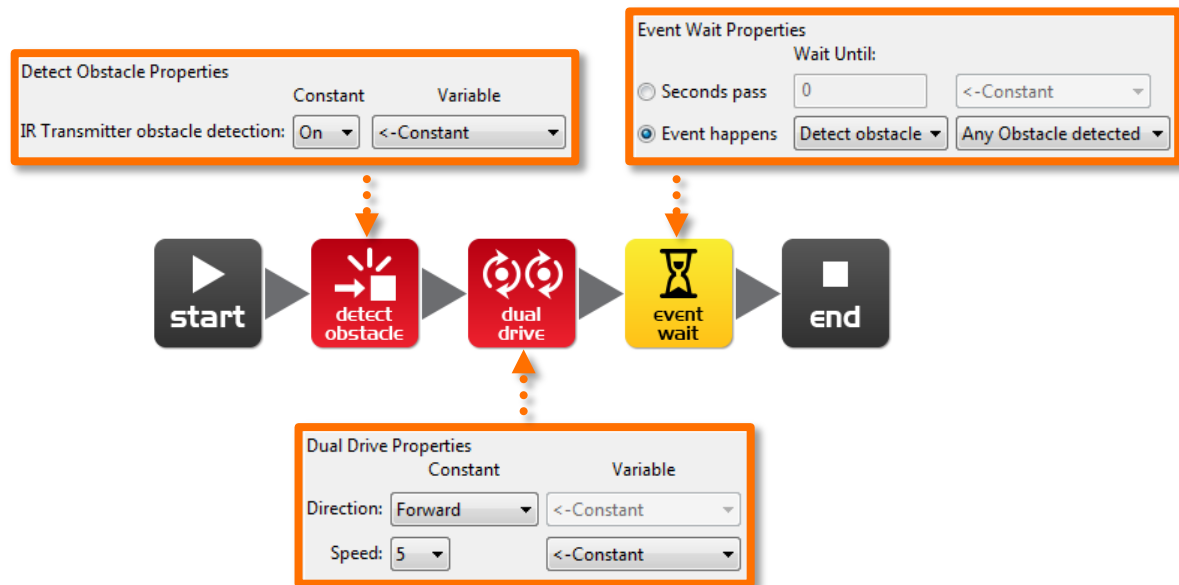
Rita in det utsända och reflekterade infraröda ljuset för detta hinder.



Namn: _____

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.



Den röda detektionsikonen slår på systemet för objekt-detektion och starta de infraröda sändarna.

Kör-ikonen 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?

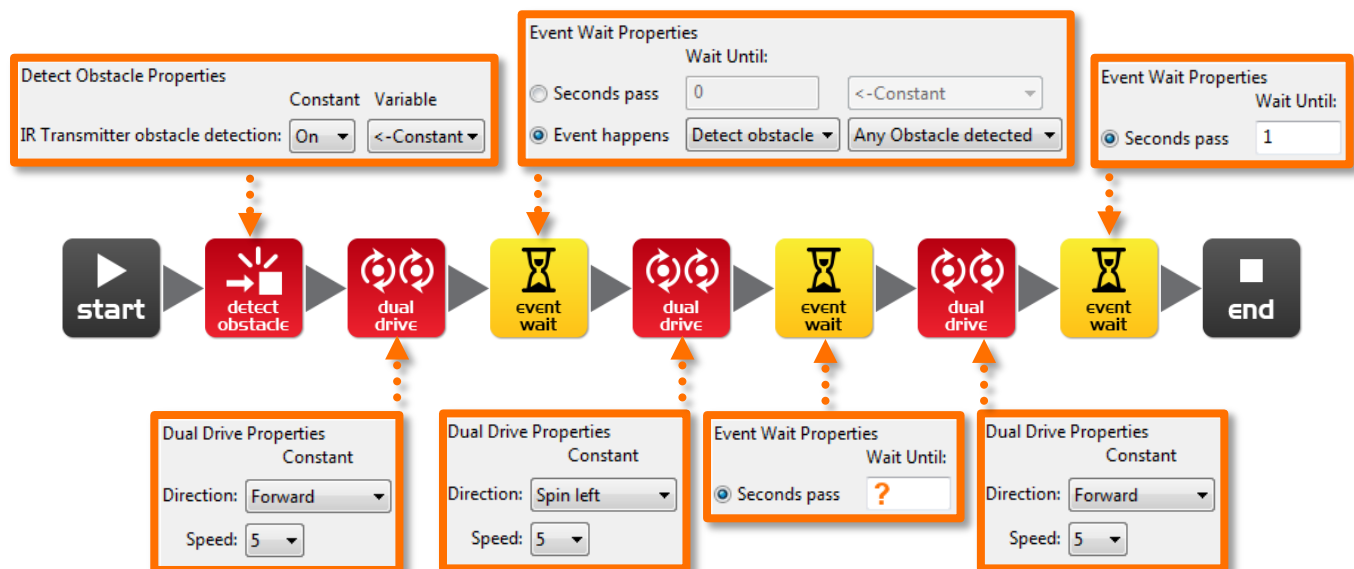
Har du sett denna typ av osynlig detektion tidigare och i så fall var?

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

Namn: _____

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 (?).

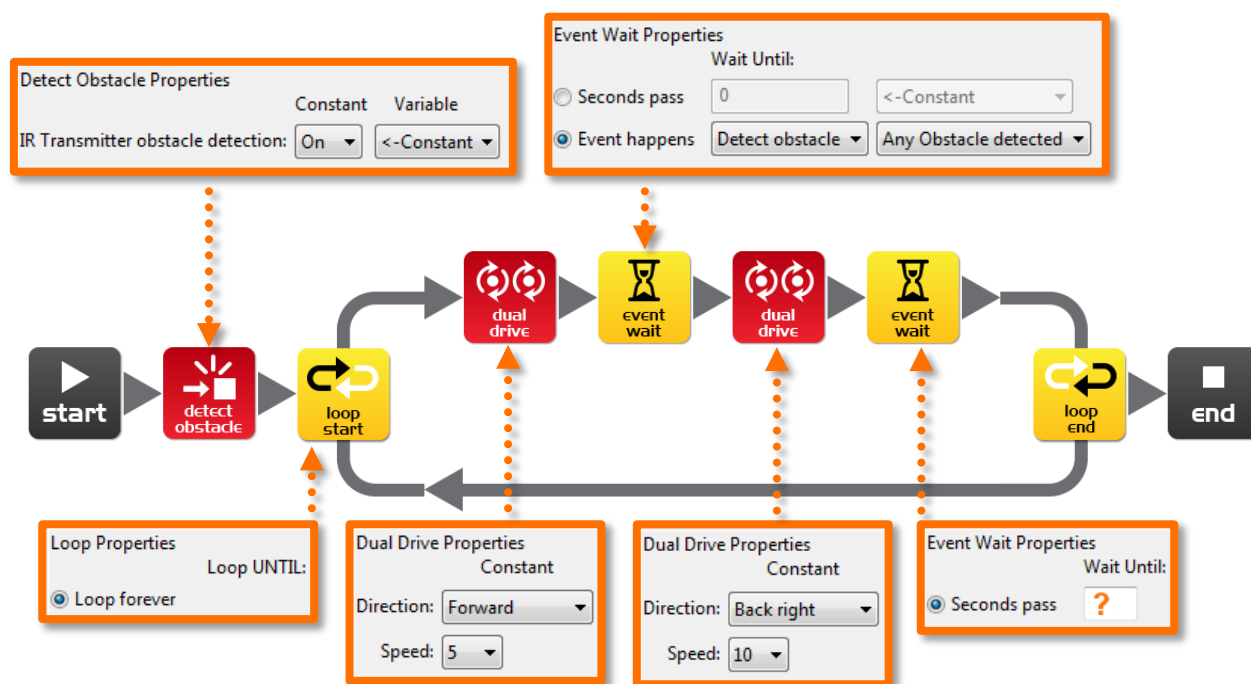
Vad är den korrekta tiden för att få din robot att svänga exakt 180°?

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

Namn: _____

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).

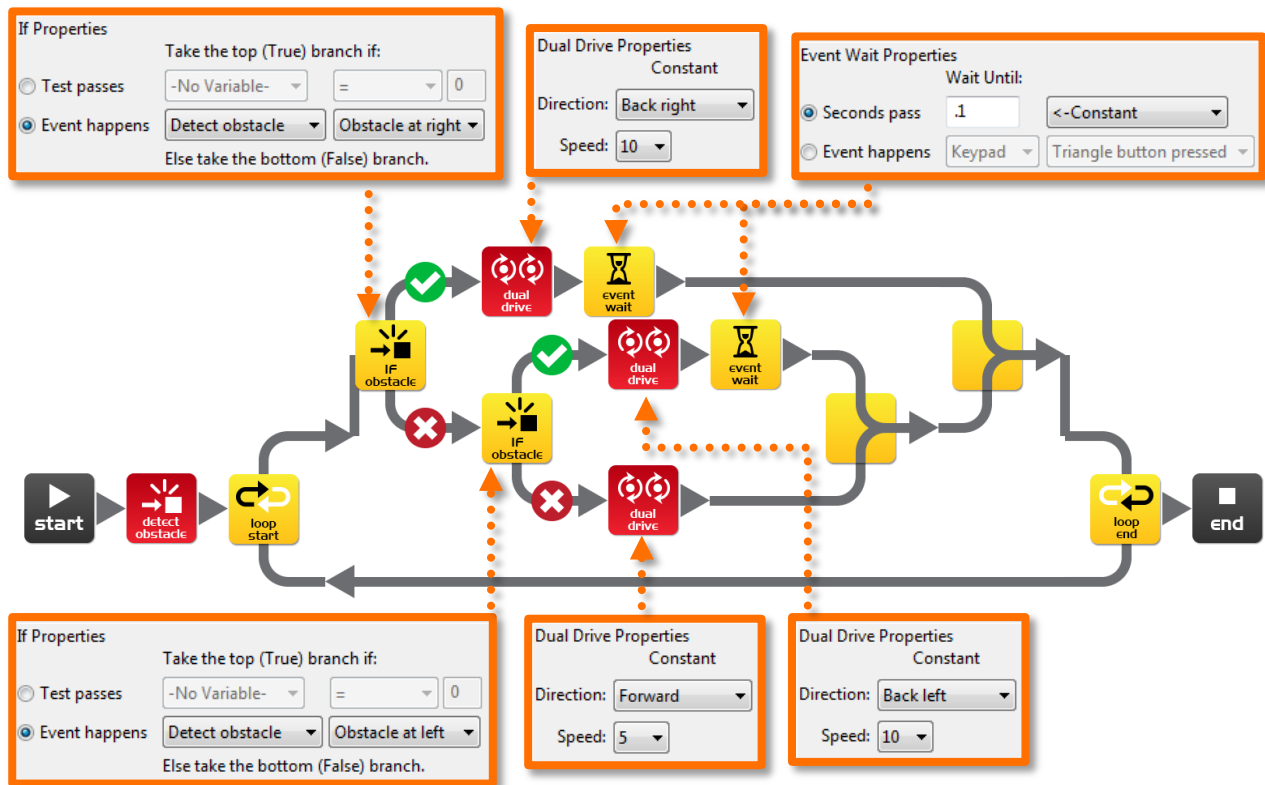
Vilken tror du är den bästa tiden för roboten att svänga på?

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

Namn: _____

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?

LEKTION 8: KÄNNA IGEN EN LINJE OCH FÖLJA DEN

Robotbeteenden för industrin – Eleverna lär sig grundläggande robotbeteenden som liknar de som används i avancerade automatiserade fabriker och lagerbyggnader.

Sensorn för linjedetektion (Arbetsblad 8.1)

Eleverna bekantar sig med robotens sensor för linjeigenkänning och hur roboten kan känna igen huruvida underlaget är vitt eller svart.

Program 1 – Kör ända till en svart linje (Arbetsblad 8.2)

Eleverna skriver ett program som kör roboten ända tills den träffar på en svart linje. De funderar även på huruvida roboten kan känna igen andra färger.

Program 2 – Håll Edison innanför området (Arbetsblad 8.3)

Eleverna skriver ett program som håller roboten innanför ett givet område. Sensorn för linjeigenkänning upptäcker områdets gräns och roboten vänder sig om. Eleverna kan använda sig av Aktivitetsblad 8.2 eller skapa sitt eget område på ett stort papper med en tjock tusch. Som rolig helklassaktivitet kan alla robotar läggas i samma inhägnade område.

Program 3 – Följa en linje (Arbetsblad 8.4)

Eleverna skriver ett program som följer en svart linje. De kan använda Aktivitetsblad 8.2 eller använda en tjock tusch (alternativt svart tejp) för att rita en egen bana på ett stort papper.

Film – Humans need not apply

Att följa en linje är ett grundläggande robotbeteende som i dag används i många automatiserade fabriker och lager. Men hur ser det ut i framtiden? Hur ser en arbetsplats med fler robotar ut i framtiden, existerar den ens? Följande film stimulerar en diskussion i klassen kring vad den enorma spridningen av robotar betyder för mänsklighetens framtid.

<https://www.youtube.com/watch?v=7Pq-S557XQU> (15 minuter)



Namn: _____

Lektion 8 Arbetsblad 8.1 – Sensor för linjeigenkänning

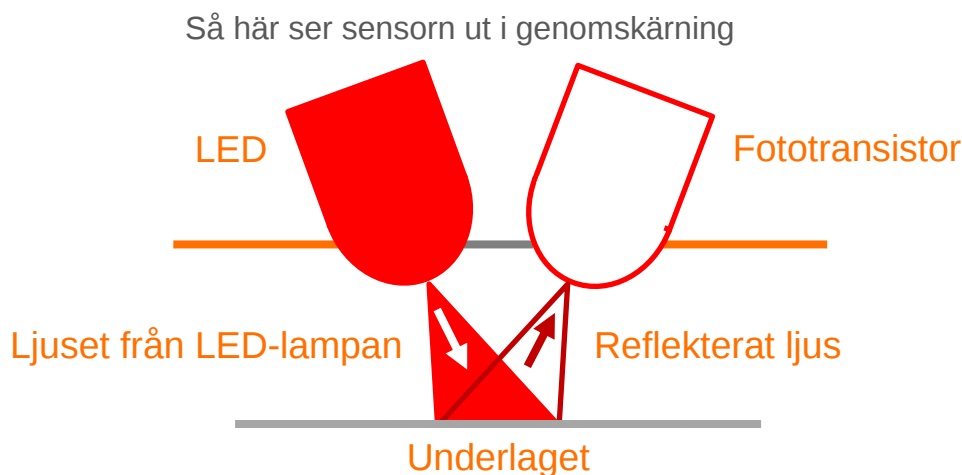
Edisonroboten är utrustad med en sensor som kan följa en linje. Sensorn består av två huvudsakliga elektroniska komponenter:

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**.

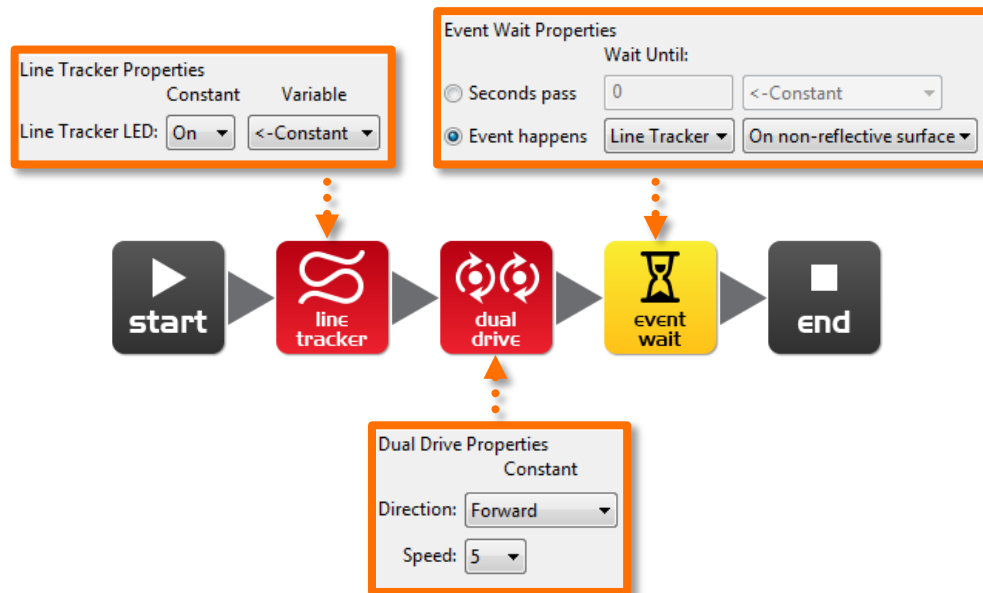
Hur tror du att linjesensorn skulle reagera på följande färgers ytor, som reflekterande eller icke-reflekterande? (Tips: Ljuset är rött)

röd yta _____, grön yta _____, blå yta _____

Namn: _____

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.

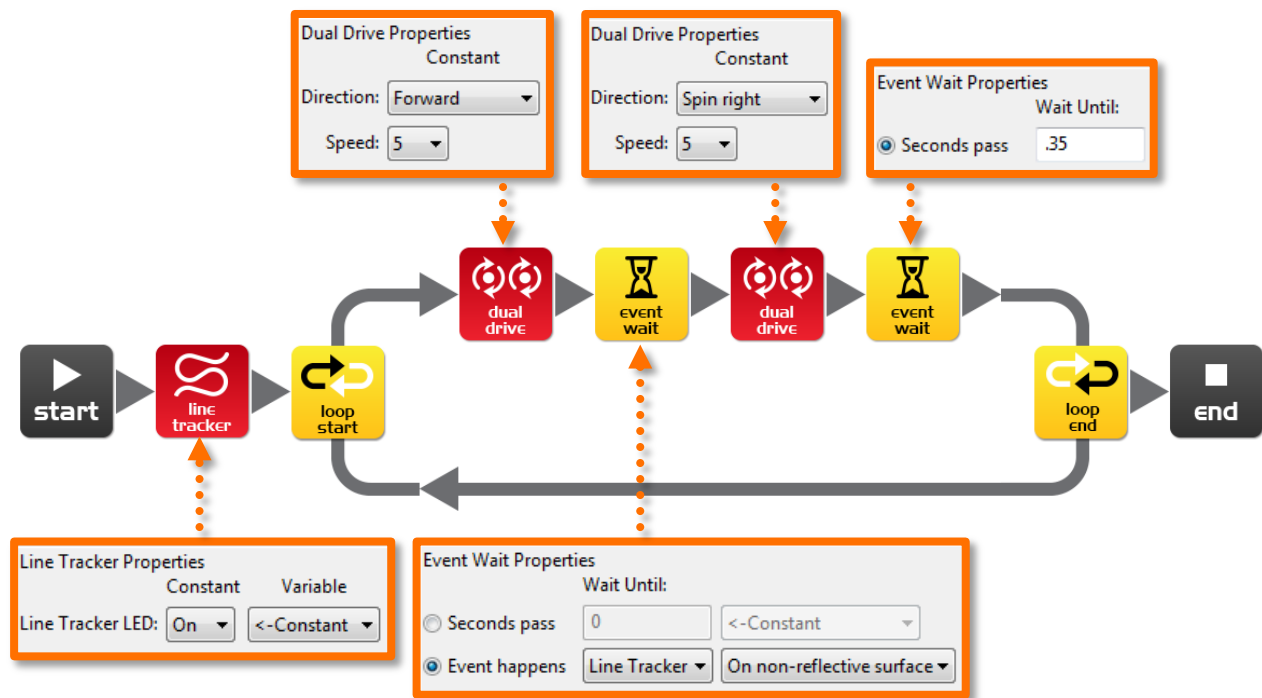
Finns det en färg som Edison inte detekterar (ser) lika bra som de andra? Vilken färg är detta? _____

Varför tror du detta händer?

Namn: _____

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.

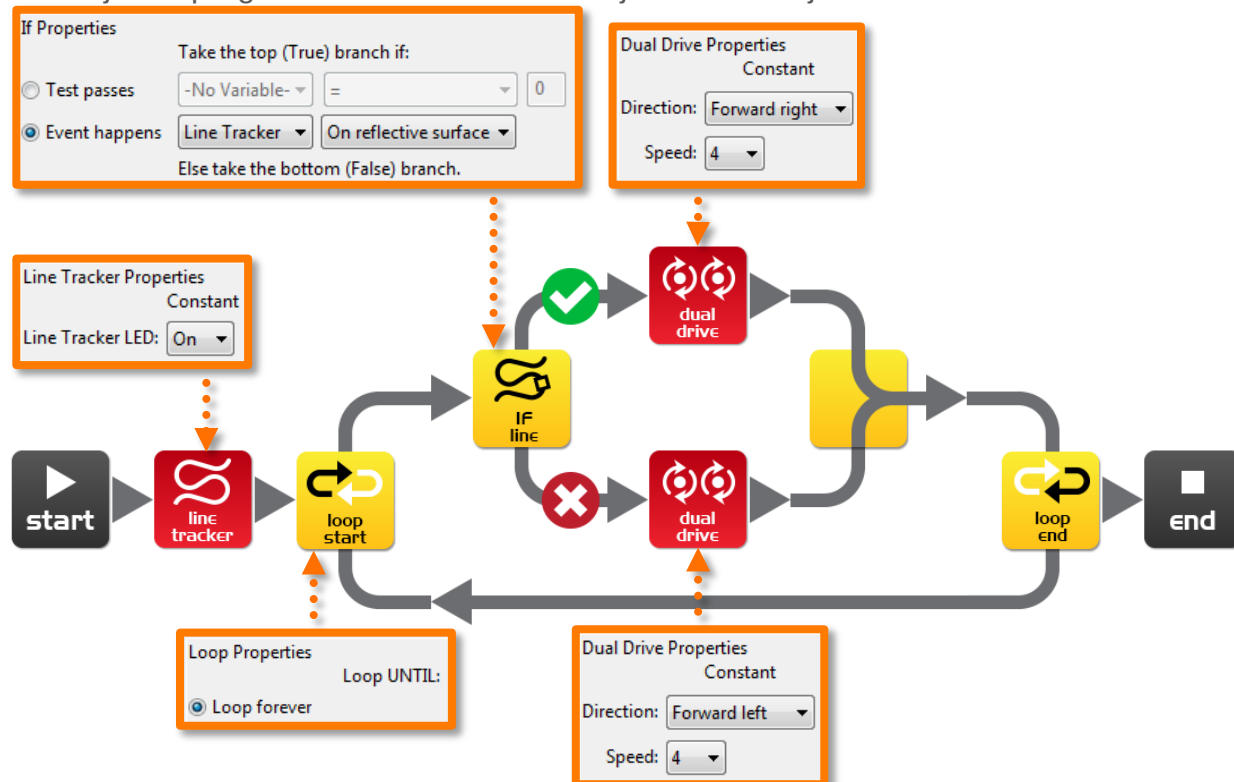
Hur snabbt kan Edison köra innan det uppstår problem?

Vad händer om Edison kör för fort?

Namn: _____

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?

Namn: _____

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?

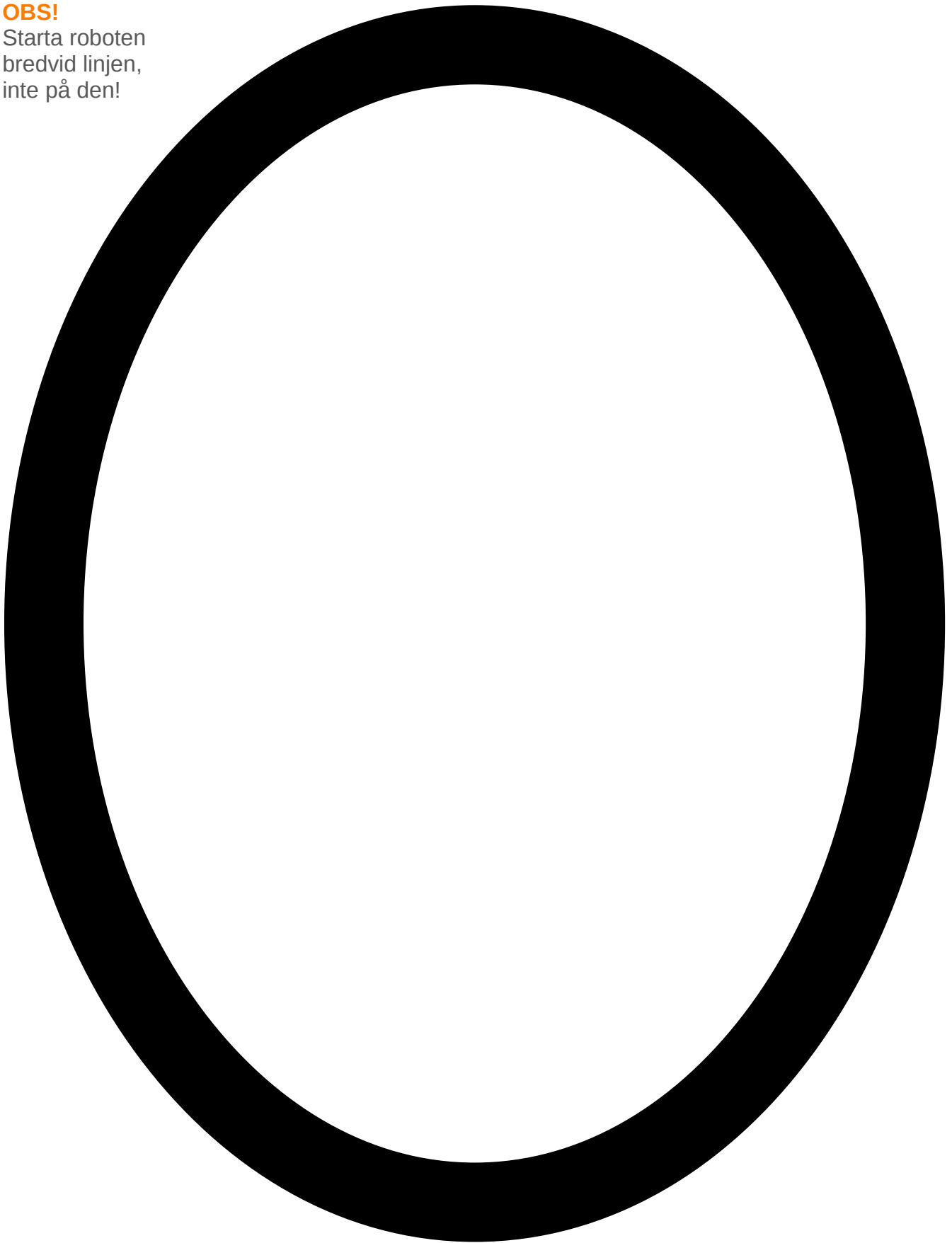


Namn: _____

Lektion 8 Aktivitetsblad 8.2 – Linjesensor

OBS!

Starta roboten
bredvid linjen,
inte på den!



LEKTION 9: REAGERA PÅ LJUS

Mätningar av miljöfaktorer och programmera matematik – Eleverna lär sig hur man mäter ljusstyrka, sparar värdena i minnet och utför matematiska kalkyler för att kontrollera robotens beteenden.

Att förstå variabler (Arbetsblad 9.1)

Eleverna lär sig vad en variabel är inom datavetenskap och hur man skapar en i Edisons mjukvara. De lär sig också om två olika storlekar på variabler; "bytes" (8 bit – 0 till 255) och "ord" (16 bit -32767 till +32767).

Program 1 – Ljusalarm (Arbetsblad 9.2)

Eleverna skriver ett program som låter som en väckarklocka när lampan i rummet slås på. Detta program introducerar användandet av variabler och använder grundläggande matematik genom "större än" (>) jämförelse.

Program 2 – Automatiska ljus (Arbetsblad 9.3)

Eleverna skriver ett program som kör framåt roboten medan den granskar ljusstyrkor. Om roboten kör in i ett mörkt rum sätter den på sina framljus. Detta program använder sig av jämförelsen "mindre än".

Program 3 – Följa ljuset (Arbetsblad 9.4)

Eleverna skriver ett program som kör roboten mot ett starkt ljussken, exempelvis en fackla/ficklampa. Detta program introducerar mer datavetenskapliga beräkningar genom att använda subtraktion och jämförelsen "mindre än".

Djuriska beteendemönster hos en robot

Programmet för att följa ett starkt ljussken demonstrerar ett robotbeteende som är väldigt likt beteendet hos en mal som attraheras av en gatulykta.

Detta kan utgöra grunden för en diskussion kring insekters och robotars intelligens.

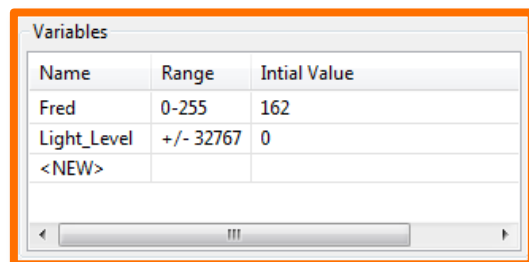
Varför är insekter levande, men inte robotar?

Namn: _____

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.



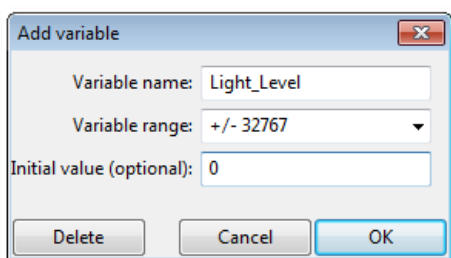
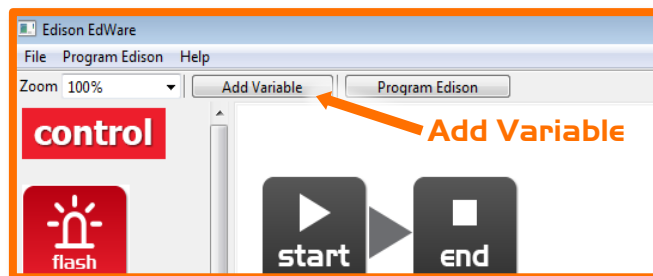
Name	Range	Initial Value
Fred	0-255	162
Light_Level	+/- 32767	0
<NEW>		

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.



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.

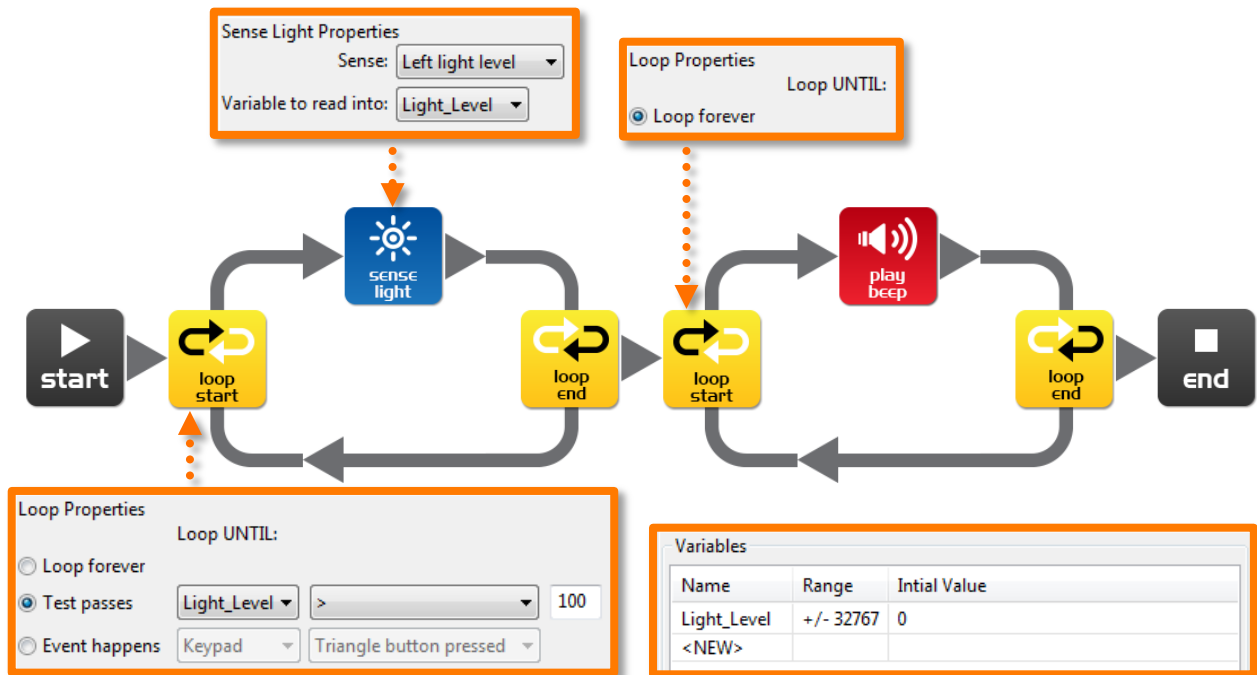
Vilken typ av variabler skulle du använda för att lagra följande tal (byte eller ord)?

12 _____, 192 _____, 801 _____, -42 _____, 27,901 _____

Namn: _____

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.

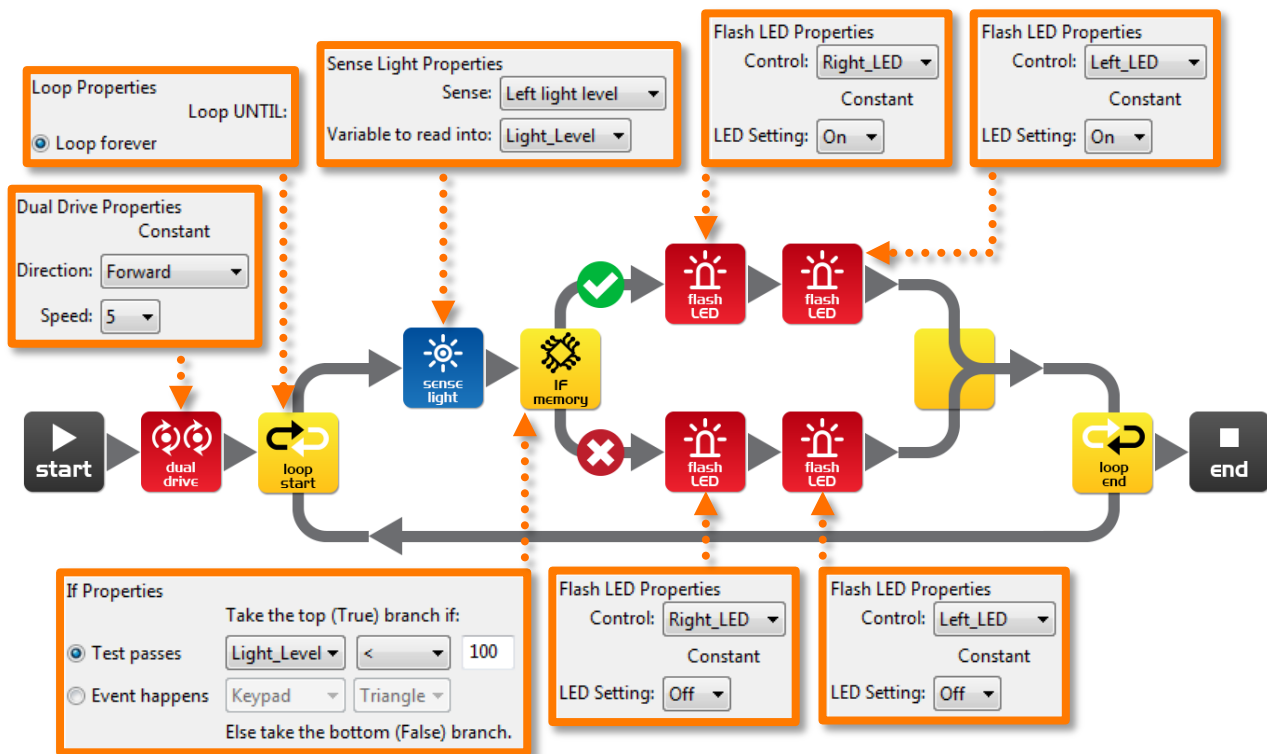
Kan du tänka dig en situation där denna typs alarm skulle kunna vara användbart?

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

Namn: _____

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 Ljusstyrka är "mindre än" 100 så tar programmet vägen med ett rättecken och slår på sina LED-lampor.

Experimentera med olika värden i OM-ikonen.

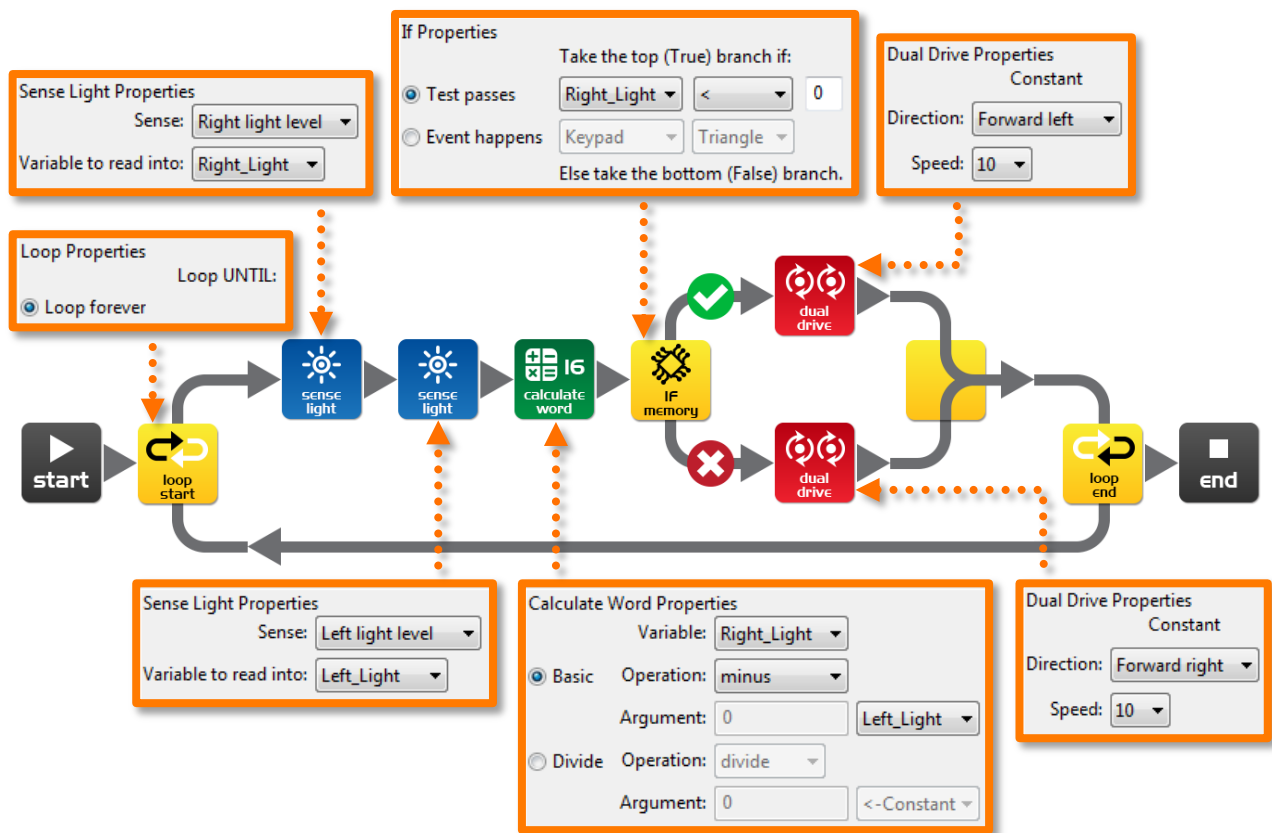
Vad händer om värdet är större än 100?

Vad händer om värdet är mindre än 100?

Namn: _____

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 (>)?

LEKTION 10: DESIGN 2 – MITT PROGRAM

Kreativt tänkande och problemlösning – Detta är den andra designen för studerande där de kan skapa egna utmaningar och fundera på hur roboten kan bidra med en lösning. Detta kan kräva två lektioners tid eller så kan det alternativt vara **hemläxa**. Eleverna kan välja ett eget tema, beskriva programmets syfte och förklara var programmet kunde användas. I denna designuppgift förväntas eleverna beskriva sin design mer detaljerat än i den första designuppgiften. Nu har de också mer kunskap om programmering och kan använda robotens sensorer.

1. Identifiera ett problem som roboten kan lösa

Eleverna har nu sett filmen *“Humans need not apply”* och är mer medvetna om hur robotar kan komma att användas i framtiden. Det här kanske stimulerar fram nya idéer från eleverna. Några exempel är:

- 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. Två robotar tävlar mot varandra i en sumobrottningsmatch.

2. Beskriv problemet eller rörelser roboten bör utföra

Eleverna **beskriver problemet** som roboten skall lösa och **hur** problemet skall lösas med hjälp av arbetsblad 10.1.

Exempel: **Problem:** “Mänskliga förare distraheras enkelt och de kan göra misstag bakom ratten. Detta misstag kan vara livshotande.” **Lösning:** Detta program demonstrerar hur en robot kan köra längs en väg och undvika att krocka med människor, andra bilar eller byggnader med hjälp av sina detektorer. Därför är denna bil mycket säkrare.’

3. Skriv programmet och testa det

Eleverna skriver sina program och testar dem.

4. Misslyckande

Det första försöket är nästan alltid misslyckat. Misslyckande är en naturlig del av programmering. Eleverna kan beskriva vad som gått fel på sitt arbetsblad.

5. Beskriv vilka ikoner som används och vad de gör

Eleverna väljer några ikoner ur sitt program som de beskriver på arbetsblad 10.2. Printa fler arbetsblad om det är önskvärt att eleverna beskriver fler av sina använda ikoner.

6. Demonstration

Eleverna demonstrerar sina program och berättar om sin idé, sina program, problemen som uppstod och hur de löste dessa.

Namn: _____

Lektion 1 O Design arbetsblad 1 O.1 – 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.

1. 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?**

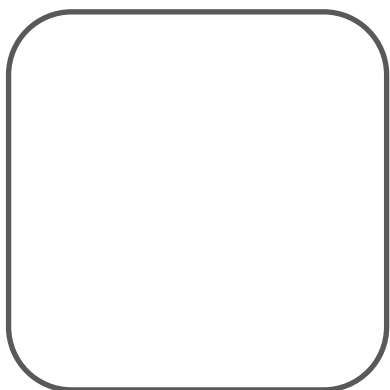
Försök igen, snart fungerar det!

Namn: _____

Lektion 10 Design arbetsblad 10.2 – Mitt program

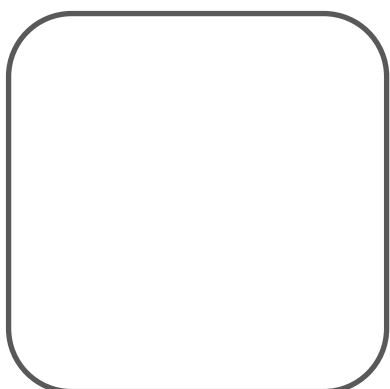
5. Beskriv några av de programmeringsikoner som du använt

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



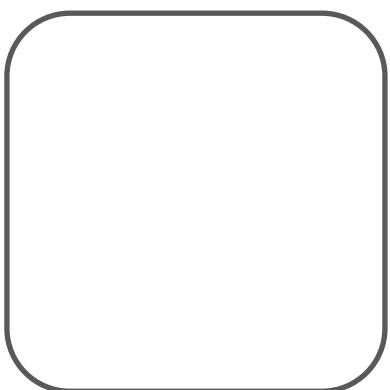
Vad heter denna ikon? _____

Vad gör denna ikon? _____



Vad heter denna ikon? _____

Vad gör denna ikon? _____



Vad heter denna ikon? _____

Vad gör denna ikon? _____

Namn: _____

Tabell över elevernas framgång

Program	Stamp
2.1 Kör roboten framåt	
2.2 Kör roboten bakåt	
2.3 Framåt och sedan bakåt	
2.4 Hastigheter	
3.1 Sväng höger	
3.2 Sväng vänster	
3.3 Sväng höger, sedan vänster	
3.4 Minilabyrint	
4.1 Utmaning	
4.2 Mexikanska vågen	
5 Design 1 – Mitt program	
6.1 Blinka med LED som reaktion på en klapp	
6.2 Kör som reaktion på en klapp	
6.3 Dansa som reaktion på en klapp	

Program	Stamp
7.1 Detektera hinder och stanna	
7.2 Detektera hinder och undvik	
7.3 Detektera hinder och undvik i en slinga	
7.4 Detektera i höger och vänster led	
8.2 Kör ända till en svart linje	
8.3 Håll Edison innanför ett område	
8.4 Följ en linje	
9.2 Ljusalarm	
9.3 Automatiska ljus	
9.4 Följ ljuset	
10 Design 2 – Mitt program	
Eget program 1.	
Eget program 2.	
Eget program 3.	

Hjärtliga gratulationer

har deltagit i
Edisons robotik- och programmeringskurs

Följande färdigheter har utvecklats:

- Design, testning och utvärdering av robotprogram med hjälp av Edisonroboten och dess programmeringsmjukvara
- Demonstration av förståelse för robotens rörelser och sensorer
- Demonstration av förståelse för principer för robotprogrammering
- Omsättning av kunskap på egna designprojekt och utmaningar
- Utvecklande av ett underbart äventyr!

