

Ontdek Edison in 10 lessen



Deze lessenserie is geproduceerd in samenwerking met www.roboticswps.com.au



Originele, Engelstalige tekst: www.meetedison.com

Lessen vertaald door Bert Grave, Zwolle 2015

Inhoudsopgave

Les 1 Werkblad 1.1 – Kennismaken met Edison	4
Les 1 Werkblad 1.2 – Programmeren met barcode	6
Les 1 Werkblad 1.3 – Ontdek EdWare	7
Les 1 Werkblad 1.4 – Testprogramma downloaden	8
Les 2 Werkblad 2.1 – Rij de robot vooruit	9
Les 2 Werkblad 2.2 - Rij de robot achteruit	10
Les 2 Werkblad 2.3 – Vooruit en dan achteruit	11
Les 2 Werkblad 2.4 – Spelen met snelheid	12
Les 2 activiteitenveld 2.1	13
Les 3 Werkblad 3.1 – Draai naar rechts over 90 graden	14
Les 3 Werkblad 3.2 – Draai naar links over 180 graden	15
Les 3 Werkblad 3.3 – Draai naar rechts, dan naar links	16
Les 3 Werkblad 3.4 – Mini doolhof	17
Les 3 Aktiviteitenveld 3.1 – Draaien	18
Les 3 Aktiviteitenveld 3.2 – Mini doolhof	19
Les 4 Werkblad 4.1 – Uitdaging	20
Les 4 Werkblad 4.2 – Mexicaanse golf	21
Les 5 Ontwerp opdracht werkblad 5.1 -Mijn programma	22
Les 5 Ontwerpopdracht werkblad 5.2 – Mijn programma	24
Les 6 Werkblad 6.1 – Knipperende LED als reactie op een klap	25
Les 6 Werkblad 6.2 – Rij als reactie op een klap	26
Les 6 Werkblad 6.3 – Dans als reactie op klappen	27
Les 7 Werkblad 7.1 – Infrarood obstakel detectie	28
Les 7 Werkblad 7.2 – Detecteer een obstakel en stop	29
Les 7 Werkblad 7.3 – Detecteer een obstakel en voorkom	30
Les 7 Werkblad 7.4 – Obstakels vermijden	31
Les 7 Werkblad 7.5 – Rechts en links detectie	32
Les 8 Werkblad 8.1 – Lijn volg sensor	34
Les 8 Werkblad 8.2 – Rij tot een zwarte lijn	35
Les 8 Werkblad 8.3 – Rij tot een grens	36
Les 8 Werkblad 8.4 – Volg een lijn	37
Les 8 Aktiviteiten blad 8.1 – Lijn volg sensor	38
Les 8 Aktiviteiten blad 8.2 – Lijn volg sensor	39
Les 9 Werkblad 9.1 – Begrijpelijke variabelen	40

Les 9 Werkblad 9.2 – Licht alarm	41
Les 9 Werkblad 9.3 – Automatische lichten	42
Les 9 Werkblad 9.4 – Licht volgen	43
Les 10 Ontwerp opdracht werkblad10.1 – Mijn programma	45
Les 10 Ontwerp opdracht werkblad 10.2 – Mijn programma	47

Les I Werkblad I.I - Kennismaken met Edison

Edison is een kleine programmeerbare robot, waar ook LEGO-onderdelen bij gebruikt kunnen worden.

Edison gebruikt sensoren en motoren. In de afbeeldingen hieronder kan je Edison's sensoren en knoppen zien.



Volg de lijn sensor = voor het volgen van een lijn: een combinatie van een LED en een lichtsensor. Met de lichtsensor kunnen ook de barcodes gelezen worden.

De kabel wordt gebruikt om de programma's van de computer/tablet over te brengen naar Edison. De stekker wordt aangesloten op de koptelefoonuitgang.

4

Installatie van de software EdWare

De EdWare programmeer software is beschikbaar voor Windows, Mac, Linux, iOS, Android en Raspberry Pi. Ga naar <u>meetedison.com/downloads</u> en kies de juiste versie en volg de aanwijzingen.

Bij Windows moet de geluidsversterking (sound enhancements) uitgeschakeld worden. In het voorbeeld hieronder de stappen voor Windows 7.



Les I Werkblad I.2 - Programmeren met barcode

Lezen van de barcode

- 1. Plaats Edison gericht naar de barcode aan de rechter kant (zie afbeelding)
- 2. Druk 3x op de recordknop (rond)
- 3. Edison rijdt vooruit en scant de barcode









Barcode - Vermijd obstakels

Barcode - Volg de lijn



Barcode – Volg het licht







Beschrijf wat er gebeurt:

Les I Werkblad I.3 - Ontdek EdWare



Dit is EdWare, de software om de Edison robot te programmeren.

Sleep iconen uit het overzicht naar het programmagebied. Plaats ze tussen de "start" en "eind" iconen. Verander de instellingen onderin het scherm. Lees ook het hulpscherm.

Wat zijn de namen of de vier hoofdgroepen van de iconen? Noteer de Engelse term en geef ook een vertaling.

Les I Werkblad I.4 - Testprogramma downloaden

Open het testprogramma (test program). Bestandnaam: Test Program.edw (Windows locatie: C:\Program Files\EdWare\My Programs)



Test program

Zo ziet een programma er uit. Edison kijkt naar één icoon tegelijk en doet precies wat dat icoon aangeeft. De pijlen geven de leesrichting van het programma aan. In dit programma is een lus ingebouwd, zodat het programma altijd doorgaat tot je het uit zet of tot de batterijen op zijn.

Het downloaden van het programma naar de Edison gaat met behulp van het kabeltje. Doe de stekker in de hoofdtelefoonuitgang en zet het volume op maximaal. De andere kant van de kabel gaat in de Edison op de manier als hiernaast in het plaatje.



Het downloaden gaat als volgt:

- 1. Druk één maal op de (ronde) opnameknop.
- 2. Klik in EdWare op 'Program Edison' en dan 'Start Download'
- 3. Druk op Edison op de (driehoekige) startknop om het programma te starten.

Wat deed de robot toen je op de startknop drukte?

Kun je een koppeling leggen tussen wat er gebeurde en de iconen in het programma? Leg uit.

Hoe werd het programma van de computer naar de robot gebracht?

Les 2 Werkblad 2. I – Rij de robot vooruit

Schrijf het volgende programma om de robot vooruit te laten rijden. Gebruik het activiteitenveld of gekleurde tape op de tafel als begin en eindpunt.

Dual Di Directio	rive Properties Constant on: Forward	▼ <-Const	/ariable ant	-	
Spe	ed: 5 🔻	<-Const	ant	-	
s tar	rt Qua dua driv		Event wait		∎ ∈nd
	Event Wait Properti	ies Wait Until:			
	Seconds pass	?	<-Const	ant	•
	Event happens	Keypad 👻	Triangle	button p	ressed 👻

Stel de "Seconds pass" (vervang dus ?) in op een aantal seconden (in tijd) die de robot nodig heeft om te starten van achter de startlijn tot de eindlijn.

Minimum hoeveelheid tijd is 0.01 seconde.

Maximum hoeveelheid tijd is 327.67 seconden.

Probeer verschillende hoeveelheden tijd tot je robot vooruit rijdt en precies stopt voor de eindlijn.

Wat is de juiste tijd om je robot van start tot eind te laten rijden?

Beschrijf wat de robot aan het doen is en waarom?

Les 2 Werkblad 2.2 - Rij de robot achteruit

Schrijf het volgende programma om de robot achteruit te laten rijden. Gebruik het activiteitenveld of gekleurde tape op de tafel als begin en eindpunt.



Stel de "Seconds pass" (vervang dus ?) in op een aantal seconden die de robot nodig heeft om te starten van achter de startlijn tot de eindlijn.

Minimum hoeveelheid tijd is 0.01 seconde.

Maximum hoeveelheid tijd is 327.67 seconden.

Probeer verschillende hoeveelheden tijd tot je robot achteruit rijdt en precies stopt voor de eindlijn.

Wat is de juiste tijd om je robot van start tot eind te laten rijden?

Beschrijf wat de robot aan het doen is en waarom?

Les 2 Werkblad 2.3 – Vooruit en dan achteruit

Schrijf het volgende programma om de robot eerst vooruit en dan achteruit te laten rijden. Gebruik het activiteitenveld of gekleurde tape op de tafel als begin en eindpunt.

Dual Drive Properties Constant Variabl Direction: Forward C-Constant Speed: 5 Constant	le T	Dual Drive Propertie Cons Direction: Backwar Speed: 5 -	s stant Variable d <-Constant
start	dual drive	dual drive	event wait
Event Wait Properties Wait Until:		Event Wait Propertie	es Wait Until:
 Seconds pass Event happens Keypad 	<-Constant ▼ Triangle button pressed ▼	Seconds pass Event happens	<-Constant

Stel de "Seconds pass" (vervang dus ?) in op een aantal seconden die de robot nodig heeft om te rijden van de startlijn tot vlak voor de eindlijn, dan achteruit te rijden en te stoppen voor de startlijn.

Minimum hoeveelheid tijd is 0.01 seconde.

Maximum hoeveelheid tijd is 327.67 seconden.

Probeer verschillende tijdshoeveelheden tot je robot vooruit en achteruit rijdt en precise voor de startlijn stopt.

Wat is de juiste hoeveelheid tijd om vooruit en achteruit te rijden?

Vooruit _____ Achteruit _____

Les 2 Werkblad 2.4 - Spelen met snelheid

Schrijf het volgende programma om de robot vooruit en achteruit te laten rijden. Gebruik het activiteitenveld of gekleurde tape lijnen op de tafel als begin en eindpunten.

Dual Drive Properties Constant Variable Direction: Forward	Dual Drive Properties Constant Variable Direction: Backward Variable
Speed: ?	▼ Speed: ? ▼ <-Constant ▼
start dual dual drive	Arent Arit
rent Wait Properties Wait Until:	Event Wait Properties Wait Until:
Seconds pass ? <-Constant Event happens Keypad Triangle button presse	Seconds pass ? <-Constant

Deze keer verandert de tijd en de snelheid!

Voel je vrij om nieuwe iconen toe te voegen aan je programma zoals bliepjes, knipperende LEDjes of misschien zelfs muziek afspelen.

1. Begin van achter de startlijn en stop voor de eindlijn, rij dan achteruit en stop voor de startlijn zo snel als mogelijk is!

Wat zijn je snelste instellingen?

Rij vooruit snelheid _____ Rij vooruit tijd _____

Rij achteruit snelheid _____ Rij achteruit tijd _____

2. Start achter de startlijn en stop voor de eindlijn,rij dan achteruit terug en stop voor de startlijn zo langzaam mogelijk!

Wat zijn je langzaamste instellingen?

Rij vooruit snelheid _____ rij vooruit tijd _____

Rij achteruit snelheid _____ rij achteruit tijd _____

Les 2 activiteitenveld 2. I





START LIJN

Les 3 Werkblad 3.1 – Draai naar rechts over 90 graden

Schrijf het volgende programma om de Edison robot 90 graden naar rechts te laten draaien (90°). Gebruik werkblad 3.1 of markeer lijnen op de tafel als begin en eindpunt.



Stel de 'Seconds pass' (dus vervang ?) op een aantal seconden in (tijd om te zorgen dat de robot precise 90 graden naar rechts draait.

Minimum hoeveelheid tijd is 0.01 seconde.

Maximum hoeveelheid tijd is 327.67 seconden.

Probeer verschillende hoeveelheden tijd tot je robot naar rechts draait van het beginpunt tot het eindpunt.

Wat is de juiste hoeveelheid tijd om je robot 90° te draaien?

Les 3 Werkblad 3.2 - Draai naar links over 180 graden

Schrijf een programma om de robot 180 graden (180°) naar links te laten draaien. Gebruik aktiviteitenveld 3.1 of lijnen op tafel als start en eindpunt.

Dual Drive Pro	perties Constant	Variable
Direction: Sp	in left 🔹 🔻	<-Constant
Speed: 5	•	<-Constant
b start	dual	event wait
	Event Wait Prop	erties Wait Until:
	Seconds pass	; ? <-Constant 🔹
	Event happer	ns Keypad 🔻 Triangle button pressed 👻

Stel de 'Seconds pass' (vervang dus ?) in op een hoeveelheid seconden zodat de robot naar links draait over precies 180°.

Minimum hoeveelheid tijd is 0.01 seconde.

Maximum hoeveelheid tijd is 327.67 seconden.

Probeer verschillende tijden uit tot je robot draait van het stratpunt tot het eindpunt.

Wat is juiste hoeveelheid tijd om je robot 180° te laten draaien?

Les 3 Werkblad 3.3 – Draai naar rechts, dan naar links

Schrijf een programma voor de robot om hem 90 graden (90°) naar rechts te laten draaien en dan 180 graden naar links (180°). Gebruik aktiviteitenveld 3.1 of gemarkeerde lijnen op tafel als begin en eindpunt.



Stel 'Seconds pass' (dus vervang?) in op een aantal seconden om de robot precies 90° naar rechts te laten draaien en dan naar links over 270° (90° + 180°).

Minimum hoeveelheid tijd is 0.01 seconde.

Maximum hoeveelheid tijd is 327.67 seconden.

Probeer verschillende hoeveelheden tijd tot je robot precies 90° naar rechts draait van af het startpunt tot het eindpunt en dan 270° naar links draait tot het onderste eindpunt.

Wat is de precieze hoeveelheid tijd om je robot 90° te laten draaien en dan 270°?

90°: _____, 270°: _____

Les 3 Werkblad 3.4 - Mini doolhof

Gebruik de programmeerkennis die je hebt opgedaan om je robot door het mini doolhof op aktiviteitenveld 3.2 te sturen

De robot moet beginnen van achter de startlijn en stoppen na de finishlijn en mag niet op of over de lijnen rijden.

Je moet meerdere programmeer iconen combineren om de robot naar voren te laten rijden en de vereiste bochten te laten maken.

Hint: vooruit, draai naar rechts, vooruit, draai naar links, vooruit

Beschrijf wat de robot doet en waarom?

Welke uitdagingen ben je tegengekomen om de robot door het doolhof te rijden ?

Moeilijker uitdaging

Wie kan de robot in de snelste tijd door het doolhof krijgen?

Het is niet nodig om een stopwatch te gebruiken . Tel gewoon de tijd van alle wacht iconen op.

Vergeet niet : De robot moet beginnen van achter de startlijn en stoppen na de finishlijn en mag niet op de lijnen rijden om te winnen .

Wat is de tijd van jouw robot om door het doolhof te rijden?

klas:

Les 3 Aktiviteitenveld 3.1 – Draaien

Plaats de robot hier beneden in de omtrek en speel dan je programma af.

Je robot moet

Programma 1 – naar rechts draaien vanaf start 0° tot einde 90°

Programma 2 - naar links draaien van start 0° tot einde 180°

Programma 3 – naar rechts draaien van start 0° tot einde 90° en dan naar links draaien over 270° tot einde 180°



Les 3 Aktiviteitenveld 3.2 – Mini doolhof



Les 4 Werkblad 4. I - Uitdaging

Kies een robot rij uitdaging voor jezelf en programmeer je Edison robot om die uitdaging te rijden.

Hier zijn enkele voorbeelden, maar je kunt ook je eigen uitdaging bedenken .

- Rijd rond obstakels, zoals een kopje of etui
- Rijd rond de rand van de tafel zonder er van af te vallen
- Maak een eigen doolhof op een groot stuk papier
- Maak een doolhof met behulp van bouwstenen zoals LEGO

Vergeet niet dat je andere programmeer iconen kunt gebruiken, zoals LED flitsen, pieptonen en muziek om zo creatief mogelijk te zijn .

Wat is jouw robot uitdaging?

Wat was moeilijk bij het schrijven van het programma?

Welke extra programmeer iconen heb je gebruikt en wat deden ze?

Les 4 Werkblad 4.2 - Mexicaanse golf

Dit is een leuke klassen activiteit, die ieders robot hetzelfde programma laat uitvoeren maar op een iets andere tijd. Het resultaat is vergelijkbaar met een Mexicaanse golf of gechoreografeerde dans.

Lussen

Gebruik een lus in je programma zodat de beweging zich herhaalt:



Wat moet je doen:

Schrijf een kort programma met 'coole' bewegingen er in van de robot. Vergeet niet om licht en geluid te gebruiken in je programma.

Wanneer je je opeenvolging van bewegingen hebt geschreven en getest kiest de leraar de beste van de klas.

Als de jouwe is gekozen dan deel je je programmabestand met de klas . Als het programma van iemand anders wordt gebruikt, moet je aan het begin van dat programma een wachten icoon toevoegen.. Stel de tijd in zoals de leraar zegt .

Onderdeel wacht tijd: _____ Robot nummer: _____ (om robots op volgorde te plaatsen)

Plaats alle robots op één lijn in volgorde van vertragingstijd waarbij de robot met de kortste tijd aan het begin en langste tijd aan het eind staat . ledereen drukt nu op hetzelfde moment op start. Dan een stapje terug doen en kijken naar de robot prestaties .

Beschrijf je *coole* robot bewegingen:

klas:

Les 5 Ontwerp opdracht werkblad 5.1 -Mijn programma

In de toekomst zijn robots onze hulpen. We hebben Roomba de robot stofzuiger al, dus die toekomst is niet ver weg. Bedenk een nuttige taak die de robot kan uitvoeren en schrijf dan een programma zodat hij die taak kan doen.

Hier zijn enkele voorbeelden:

- Dansen op muziek een entertain robot
- Stofzuiger erg handig in het huis
- Bewakingsrobot je etui laten bewaken

I. Discusseer

Naam:

Bespreek dit met de leerling naast je of met andere leerlingen om te beslissen over een bruikbare robot toepassing die je kunt programmeren.

2. Beschrijf

Voordat je begint met het schrijven van het programma moet je eerst hieronder het probleem dat je robot gaat oplossen beschrijven en hoe hij dat gaat doen.

Het probleem is:

Mijn robot lost dit op door...

3. Schrijf je programma

Plan je programma voordat je begint met iconen op zijn plaats te zetten. Gebruik de iconen die je hebt geleerd maar je kunt ook proberen andere iconen te gebruiken.

4. Falen?

Niet alles werkt de eerste keer op de manier waarop je denkt. Geen probleem! Probeer het opnieuw maar beschrijf eerst wat er verkeerd ging. Waarom werkte je programma de eerste keer niet? Denk aan dit: de beroemde Thomas Edison faalde 10.000 keer voordat hij de succesvolle gloeilamp uitvond! Dus *blijf proberen!*

Les 5 Ontwerpopdracht werkblad 5.2 - Mijn programma

5. Beschrijf een aantal iconen uit je programma

Teken en kleur ze in de vakken hieronder. Dan beschrijf je wat ze doen in je programma.

Wat is de naam van dit icoon?
Wat doet dit icoon?
Wat is de naam van dit icoon?
Wat doet dit icoon?
Wat is de naam vandit icoon?
Wat doet dit icoon?

Les 6 Werkblad 6. I – Knipperende LED als reactie op een klap

Schrijf het volgende programma om de robot het LED te laten knipperen als antwoord op een hard geluid zoals een klap..



In dit programma is het eerste wacht icoon niet gebruikt om een tijdsperiode te wachten. In plaats daarvan zal het programma niet verdergaan vanaf dit punt tot er een speciale gebeurtenis plaatsvindt.

Binnen de eerste wachten icoon selecteer je 'Event happens'(er gebeurt iets) en selecteer de type gebeurtenis die je denkt dat nodig is om te reageren op een klap.

Op welke afstand tot je robot reageert je robot nog op je klap?

Wat is het doel van de lus iconen in dit programma en wat er zou gebeuren als de lus iconen er niet waren?

Les 6 Werkblad 6.2 – Rij als reactie op een klap

Schrijf het volgende programma om de robot vooruit te laten rijden als antwoord op een klap.



De geluidssensor van de Edison robot is niet alleen gevoelig voor klappen, maar kan ook reageren op een hard geluid. Dit omvat de robot motoren, tandwielen en wielen, die ook geluid maken als ze draaien. Om te voorkomen dat dit geluid van de robot de inwerkingtreding van de geluidsensor veroorzaakt zijn er twee extra iconen in dit programma opgenomen en zijn hierboven gemarkeerd met oranje sterren. Het eerste wachten icoon is ingesteld op 0,1 seconde en geeft de robotmotoren tijd om te stoppen. Het tweede wachtenicoon is ingesteld om te wachten op een klap en zal het programma dan door laten gaan, omdat het geluid van de motor zal zijn gedetecteerd. Je moet deze twee iconen gebruiken na het stoppen van de motoren als je de geluidssensor wilt gebruiken.

Experimenteer met verschillende richtingen in het eerste icoon van dubbele aandrijving en met de 'volgende gebeurtenis wachten' icoon om verschillende bewegingen maken.

Welke andere richtingen en tijden heb je geprobeerd?

Experiment 1: richting	, tijd
Experiment 2: richting	, tijd
Experiment 3: richting	, tijd

Les 6 Werkblad 6.3 - Dans als reactie op klappen

Gebruik wat je hebt geleerd in de voorgande twee oefeningen om een dansroutine te ontwerpen waarin je robot reageert op je klappen.



Je hebt ten minste twee dans manoeuvres nodig, maar je kunt er zoveel als je wilt toevoegen. Het bovenstaande programma heeft twee enkele actie manoeuvres. Deze twee manoeuvres herhalen zich omdat ze in een lus staan.

Jij kunt dit ook proberen en voeg dan twee manoeuvres per klap toe..

Alternatieve uitdaging

Als je geen dansprogramma wilt ontwerpen, dan kun je de robot een korte hindernisbaan laten rijden als reactie op een klap en laat je de robot omkeren en teruggaan naar aanleiding van een tweede klap.

Hoeveel manoeuvers heeft jouw programma?

Beschrijf je robots dans of rij manoeuvres.

Leuke klassen aktiviteit

Selecteer de beste dans routine van de klas en laat iedere leerling dit programma gebruiken. Zet dan de robots bij elkaar en iedereen klapt op tijd in zijn handen om de robots te zien dansen.

Les 7 Werkblad 7.1 - Infrarood obstakel detectie

De Edison robot is uitgerust met een infrarood (IR) "zicht". IR licht is niet zichtbaar voor het menselijk oog, dus je kunt dit licht niet zien, maar het laat de Edison "zien" in het donker.

Om obstakels te detecteren zendt de Edison zendt IR-licht uit aan de voorzijde en aan de linker als de rechterzijde van de robot. Wanneer het IR-licht wordt gereflecteerd door een obstakel, zoals een muur, dan wordt het gereflecteerde licht gedetecteerd door de Edison door één IR detector. De IR detector aan de voorkant van de robot.

In de onderstaande afbeelding is er een obstakel linksvoor de Edison, dus alleen infrarood licht van de linker zender wordt gereflecteerd. Door het ontvangen signaal kan Edison bepalen dat er een obstakel is aan de linkerkant, maar dat er rechts geen belemmering is.

Uitgezonden IR-licht wordt weergegeven in rood en gereflecteerd infrarood licht wordt weergegeven in blauw.



Teken het uitgezonden IR licht en het gereflecteerde IR licht bij dit obstakel.



Les 7 Werkblad 7.2 – Detecteer een obstakel en stop

Schrijf het volgende programma om de Edison robot dte laten rijden totdat het een obstakel tegenkomt.

Detect Obstacle Properties	Event Wait Propert	ies Wait Until:	
Constant Variable	Seconds pass	0	<-Constant 👻
IR Transmitter obstacle detection: On 🔹 <-Constant 💌	Event happens	Detect obstacle 🔻	Any Obstacle detected 💌
start	event wait	€nd	
Dual Drive Properties Constant Va	ariable		
Direction: Forward	int 👻		
Speed: 5 -Consta	int 🔻		

De rode (controle groep) detecteer obstakel icoon is nodig om het obstakel detectie systeem aan te zetten en start de IR-licht uitzenders.

De tweevoudige rijsnelheid is ingesteld op 5 om de robot om een obstakel te laten te rijden voordat hij ermee botst. Als de snelheid te hoog is dan zal de robot botsen met obstakels.

Vanaf welke afstand kan de robot obstakels detecteren?

Heb je deze vorm van onzichtbare detective wel eens eerder gezien en waar?

Waar denk je dat dit sort van detective technologie ook gebruikt kan worden?

Les 7 Werkblad 7.3 – Detecteer een obstakel en voorkom

Schrijf het volgende programma om de robot te laten rijden tot hij een obstakel tegenkomt, dan moet hij 180° draaien en gedurende 1 seconde weg rijden van het obstakel.



In les 3 (Werkblad 3.2) heb je de juiste tijd gevonden om de robot 180° te laten draaien. Gebruik deze tijd weer op de plek die hierboven aangegeven is door het oranje vraagteken (?).

Wat is de juiste hoeveelheid tijd om je robot 180° te laten draaien?

Waarom is dit programma onvolledig denk je en hoe kun je dat verbeteren?

Les 7 Werkblad 7.4 - Obstakels vermijden

Schrijf het volgende programma om de Edison robot steeds de obstakels te laten vermijden.



Experimenteer met verschillende hoeveelheden tijd in het tweede wacht icoon. Deze tijdsinstelling bepaalt hoe lang de robot draait en wordt hierboven aangegeven door het oranje vraagteken (?).

Wat denk je dat de beste tijdsinstelling is voor de robot om te draaien?

Waarom is deze tijdsinstelling het beste? Wat doet de robot anders zodat het beter gaat?

Les 7 Werkblad 7.5 - Rechts en links detectie

Schrijf het volgende pragramma om de Edison steeds links en rechts obstakels te laten vermijden.



In het bovenstaande programma hebben we 'IF' iconen gebruikt. Dit zijn erg belangrijke iconen, omdat ze de robot de mogelijkheid geven om beslissingen te nemen zonder menselijk ingrijpen. Wanneer dit voorkomt bij een robot dan noemen we dat tegenwoordig een autonome robot, hij heeft kunstmatige intelligentie..

Een 'lf' icoon vraagt of een voorwaarde waar is of niet (true or false). Als het resultaat waar is dan neemt het programma het pad met het vinkje. Als het resultaat niet waar is neemt het programma het pad met het kruisje.

Het bovenstaande programma heeft drie verschillende paden die het kan nemen op basis van waar een obstakel. Verklaar in je eigen woorden wat de drie paden je robot laten doen.

Geen obstakel gedetecteerd:

Obstakel gedetecteerd aan de rechterkant:

Obstakel gedetecteerd aan de linkerkant:

Is de robot levend omdat het beslissingen kan nemen!? Waarom denk je dat?

Les 8 Werkblad 8.1 – Lijn volg sensor

De Edison robot is uitgerust met een lijn volg sensor. De sensor is gemaakt van twee elektronische componenten:

- 1. Rode Light Emitting Diode (LED)
- 2. Fototransistor (licht sensor)

De LED schijnt licht op het oppervlak waar de robot op rijdt. Als je twee keer op de ronde knop op de Edison drukt gaat de LED branden. Als je de robot iets optilt kun je van vlakbij de ronde lichtvlek zien dat de LED produceert op het oppervlak. Vergelijk de helderheid van het licht als de robot op een zwarte en een witte ondergrond staat.

Is het licht helderder (reflecteert meer licht) op een zwarte of witte ondergrond?

De fototransistor is een lichtsensor en meet de hoeveelheid licht dat is gereflecteerd van het oppervlak.



Dwarsdoorsnede van de lijn volgen sensor

Zoals je zag in de bovenstaande oefening, wordt er meer licht gereflecteerd door een wit oppervlak dan door een zwart oppervlak. De fototransistor ziet meer licht op een witte ondergrond dan op een zwarte ondergrond. Hierdoor kan de robot geprogrammeerd worden en reageren op het oppervlak waar het op rijdt. Een zwart oppervlak wordt geacht om 'niet-reflectief' te zijn en een wit oppervlak wordt geacht om 'reflectief' te zijn.

Hoe denk je dat de lijn volger zal reageren op de volgende oppervlaktekleuren, reflectief of niet-reflectief? (Hint: het licht is rood)

rode oppervlakte _____, groene oppervlakte _____,

blauwe oppervlakte_____

Les 8 Werkblad 8.2 - Rij tot een zwarte lijn

Schrijf het volgende programma om de Edison robot over een wit oppeervlak te laten rijden (reflectief) tot een zwarte lijn (niet-reflectief) gepasseerd wordt..



Om de lijn volg sensor te gebruiken in je programma moet je eerst de sensor aan zetten. Dit activeert ook de rode LED.

Gebruik de zwarte lijn op Aktiviteitenblad 8.1 of teken een zwarte lijn op een stuk papier. Je kunt ook zwarte elektriciteits tape op een witte tafel gebruiken. Rijdt de robot richting de zwarte lijn en hij zal stoppen.

Op Aktiviteitenblad 8.1 zijn ook gekleurde lijnen. Deze zijn rood, blauw en groen. Rijdt de Edison in de richting van elke lijn en kijk of hij stopt.

Is er een kleur dat de Edison niet goed kan detecteren (zien)? Welke kleur is dat?

Wat denk je dat er gebeurt is?

Les 8 Werkblad 8.3 – Rij tot een grens



Schrijf het volgende programma om de Edison robot tot een grens te laten rijden.

Gebruik Aktiviteitenblad 8.2 als grens of maak je eigen door een groot stuk papier te gebruiken en een dikke zwarte stift

Je kunt ook zwart elektriciteits tape op een witte tafel gebruiken als grens.

Als je een grote grens maakt kijk dan eens met heel veel robots wat er gebeurt.

Experimenteer verder met verschillende snelheden.

Hoe snel kan de robot rijden voordat er problemen komen?

Wat gebeurt er als de robot te hard rijdt?

Les 8 Werkblad 8.4 - Volg een lijn

Schrijf het volgende programma om de Edison robot een zwarte lijn te laten volgen.



Dit programma gebruikt het 'IF' icoon om de robot te laten beslissen wat te doen. Als de lijn volg sensor op een reflectieve oppervlakte (wit) is, rij dan met snelheid 4 vooruit naar rechts. Als de lijn volg sensor niet op een reflecterend oppervlak is rij dan met snelheid 4 vooruit naar links. Als de robot op wit rijdt blijft het op wit. Als de robot op zwart rijdt blijft het op zwart. Doordat hij dit doet gaat hij vooruit en volgt de rand van de lijn.

Plaats de robot binnen het vak op Aktiviteitenblad 8.2 en kijk hoe de robot de lijn volgt. Welke kant rijdt hij op (met de klok mee of er tegenin)?

Plaats de robot buiten het vak op Aktiviteitenblad 8.2 en kijk hoe hij de lijn volgt. Welke kant rijdt hij op (met de klok mee of er tegenin)?

Waarom rijdt de robot niet in dezelfde richting rond denk je?

Les 8 Aktiviteiten blad 8.1 – Lijn volg sensor

Gebruik dit Aktiviteiten blad om je programma van werkblad 8.2 te testen.

Je kunt ook je antwoorden controleren van werkblad 8.1 door de gekleurde lijnen hieronder te gebruiken. Voor welke kleuren stopt de Edison?





Les 8 Aktiviteiten blad 8.2 - Lijn volg sensor



Les 9 Werkblad 9.1 - Begrijpelijke variabelen

Een variabele is een stukje computergeheugen voor gegevensopslag. Wat variabelen zo nuttig maakt is dat deze gegevens kunnen wijzigen terwijl het programma draait, vandaar de naam variabele.

Variabelen houden van getallen zoals 10, 106, 1,482 etc.en laat een computerprogramma de wiskunde doen, dit is iets waar computers erg goed in zijn.

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

Program Edison

end

dd Variable

De Edison robot heeft twee soorten variabelen die *'bytes'* en *'words'* worden genoemd. *Byte* variabelen zijn getallen die variëren van 0 tot 255. *Word* variabelen kunnen getallen die variëren van -32767 tot 32.767 opslaan.

Om variabelen makkelijk te kunnen gebruiken geven we ze namen. Dit helpt ons mensen onthouden wat voor soort informatie is opgeslagen in hen. In EdWare, kun je je variabelen noemen net zoals je wilt. Je kunt er een '*Fred*' noemen, maar dat helpt niet erg om te onthouden welke informatie Fred voorstelt. Een betere naam kan zijn '*Licht_niveau*'. Dit soort namen maken het erg makkelijk om te onthouden waar de variabele voor gebruikt wordt en welk

> Edison EdWare File Program Edison Help

contro

Zoom 100%

soort informatie er te vinden is.

Nu je iets van variabelen weet kun je er een maken in Edware bij de volgende programma's.

Klik op de 'Add Variable' (toevoegen variabele) knop boven in de linkerhoek en een pop-up schem zal verschijnen.

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

Typ de naam van je variabele in zoals 'Licht_Niveau', selecteer het bereik van de variabele als +/-32767 (word) en zet de startwaarde (initial value) op nul. Klik nu op OK en je variabele zal worden toegevoegd aan de tabel met variabelen in de rechterbenedenhoek

Welk type variabelen zou je gebruiken voor de

Add Variable

start

volgende getallen (byte or word)?

12,	192,	801	-42	, 27,901

Les 9 Werkblad 9.2 – Licht alarm

Schrijf het volgende programma om de robot een alarmgeluid te laten maken als de lichten in de kamer ingeschakeld zijn.



Het licht sensor icoon leest de hoeveelheid licht van de linker lichtsensor en zet deze waarde in de variabele die hier 'Light_Level' heet.

De eerste lus gebruikt wiskunde om te kijken wat er moet gebeuren:

Blijf in de lus tot de waarde van de variabele 'Light_Level' groter is dan (greater than) (>) 100.

Wanneer de waarde in Light_Level groter is dan 100 wordt de lus verlaten en ga je naar de volgende lus die het alarm laat afgaan.

Zet de Edison robot in het donker en druk op de start knop. Als de lichten aan gaan slaat de robot alarm.

Kun je iets bedenken uit het echte leven waar dit handig voor is?

Welke verandering heb moet er gemaakt worden in het programma om het een donker alarm van te maken

Les 9 Werkblad 9.3 – Automatische lichten

Schrijf het volgende programma om de robot de twee LED lichten aan te laten doen als het donker wordt.



Rij de robot door een tunnel of onder een kussen en zie dat de voorlichten aan gaan.

In dit programma gebruiken we het 'less than' (<) symbool om het pad van het programma te bepalen. Als de variabele Light_Level minder is ('less than') dan 100, neem dan het ware pad en zet de LED's aan.

Experimenteer met de waarde (100) in de If icoon.

Wat gebeurt er als je het hoger maakt?

Wat gebeurt er als je het lager maakt?

Les 9 Werkblad 9.4 – Licht volgen

Schrijf het volgende programma om de Edison robot het licht van een lampje/zaklantaarn te laten volgen.

Sense Light Properties Sense: Right light level 💌 Variable to read into: Right_Light 💌	If Properties Take Test passes Event happens Else t	the top (True) branch if: t_Light • < • 0 pad • Triangle • ake the bottom (False) branch.	Dual Drive Pro Direction: Fo Speed: 10	operties Constant arward left
Loop Properties Loop UNTIL:	Sense iight	Word Properties		Vop end Dual Drive Properties Constant
Variable to read into: Left_Ligh	Basic Divide	Operation: minus Argument: 0 e Operation: divide Argument: 0	Left_Light ▼ <-Constant ▼	Direction: Forward right Speed: 10

Schijn met een lampje/zaklantaarn op Edison en de robot zal naar het licht rijden.Het programma voert een berekening uit met twee variabelen. In het Calculate Word icoon wordt de variabele Right_Light afgetrokken van de variabele Left_Light. Het resultaat (antwoord) wordt teruggeplaatst in de variabele Right_Light. Laten we eens kijken hoe het resultaat er uit zou zien:

	Right_Light	Left_Light	Calculation	Result
Lamp rechts	200	100	200 - 100 =	-100
Lamp links	100	200	100 - 200 =	+100

In het bovenstaande kun je zien dat als de lamp aan de rechterkant is het resultaat onder nul is (negatief getal). Wanneer de lamp aan de linkerkant is is het resultaat boven nul (positief getal). Het lf icoon vraagt: *Is het resultaat minder dan nul?* Als het waar is rijdt de robot naar links (richting het licht), als het niet waar is rijdt de robot naar rechts (richting het licht).

Wat zal er gebeuren als je het kleiner dan symbool (<) in een groter dan symbool (>) verandert?

Les IO Ontwerp opdracht werkblad IO.I – Mijn programma

De video 'Humans need not apply' geeft inzicht in hoe robots kunnen worden toegepast in de toekomst. Denk aan een nuttige taak die de robot kan uitvoeren en schrijf dan een programma die de robot die taak laat uitvoeren.

Hier zijn een paar voorbeelden:

- Reddende robot De robot rijdt binnen een grens op zoek naar een verdwenen person (mini figuur of pop). Als de robot de persoon lokaliseert gaat het alarm af.
- Chauffeurloze auto De robot rijdt op een aangewezen weg (lijn) zonder mensen, andere auto's of gebouwen (speelgoed) te raken.
- Vecht robots Stuur robots naar oorlogsgebieden in plaats van mensen.
 Maak een sumo ring en programmer twee of meer robots om naar elkaar te kijken en dan te proberen de andere uit de ring te duwen.

I . Discussie

Deze taak bespreek je met die naast je zit of met andere leerlingen om te beslissen welke betekenisvolle robotapplicatie je kunt programmeren.

Welke ideeën kwamen bij je op?

Wat was de reden dat een van de ideeën niet haalbaar was?

N	2	2	n	h		
IN	a	a		ł	•	_

2. Beschrijf

Voordat je je programma gaat schrijven moet je eerst het probleem beschrijven dat je robot gaat oplossen en hoe hij het gaat oplossen (de oplossing).

Probleem:

Oplossing:

3. Schrijf je programma

Plan je programma voordat je begint met iconen op hun plaats te zetten. Gebruik de iconen die je hebt geleerd te gebruiken, maar je kunt ook andere iconen proberen.

4. Mislukt?

Niet alles werkt meteen zoals je denkt dat het zal gaan. Geen probleem! Probeer het opnieuw, maar beschrijf eerst wat er mis ging. Waarom werkte je programma niet de eerste keer? Onthoud dit: Thomas Edison beroemde faalde 10.000 keer voordat hij met succes de uitvinder werd van de gloeilamp! Dus, blijf proberen!

Les IO Ontwerp opdracht werkblad IO.2 - Mijn programma

5. Beschrijf een aantal van je programmeericonen uit je programma

Teken en kleur de iconen hieronder. Beschrijf dan wat ze doen in je programma.

Wat is de naam van dit icoon?
Wat doet dit icon?
Wat is de naam van dit icoon?
Wat doet dit icoon?
Wat is de naam van dit icoon?
Wat doet dit icoon?

Naam: _____klas:_____

Leerling volg kaart

Programma	OK?	Prog
2.1 Rij de robot vooruit		7.2 E stop
2.2 Rij de robot achteruit		7.3 E voorl
2.3 Vooruit en achteruit		7.4 (
2.4 Spelen met snelheid		7.5 F
3.1 Rechts draaien		8.2 F
3.2 Links draaien		8.3 F
3.3 Draai rechtsom dan linksom		8.4 \
3.4 Mini doolhof		9.2 L
4.1 Uitdaging		9.3 A
4.2 Mexicaanse golf		9.4 L
5 Ontwerpopdracht 1 – Mijn programma		10 C prog
6.1 Knipperende LED als reactie op klap		Eige
6.2 Rijden als reactie op klap		Eige
6.3 Dansen als reactie op klappen		Eige

Programma	OK5
7.2 Detecteer een obstakel en stop	
7.3 Detecteer een obstakel en voorkom	
7.4 Obstakels vermijden	
7.5 Rechts en links detectie	
8.2 Rij tot een zwarte lijn	
8.3 Rij tot een grens	
8.4 Volg een lijn	
9.2 Licht alarm	
9.3 Automatische lichten	
9.4 Licht volgen	
10 Ontwerpopdracht 2 – Mijn programma	
Eigen programma 1.	
Eigen programma 2.	
Eigen programma 3.	





Gefeliciteerd!

Heeft succesvol succesvol afgerond de

Edison robotica en programmeer cursus

De volgende vaardigheden zijn behaald:

- Ontwerpen, testen en evalueren van robotprogramma's door de Edison robot en zijn programmeersoftware te gebruiken
- Gedemonstreerde bewegingen en sensoren van de robot begrijpen
- Gedemonstreerde robotprogramma principes begrijpen
- Toepassen van het geleerde in korte ontwerp opdrachten
- Een geweldig gevoel voor avontuur hebben!

