

# Docentenhandleiding



Auteur	Physical Computing
Laatst gewijzigd	17 july 2019
Licentie	CC Naamsvermelding 4.0 Internationale licentie
Webadres	<a href="https://maken.wikiwijs.nl/136757">https://maken.wikiwijs.nl/136757</a>



Dit lesmateriaal is gemaakt met Wikiwijs van Kennisnet. Wikiwijs is hét onderwijsplatform waar je leermiddelen zoekt, maakt en deelt.

# Inhoudsopgave

Inleiding

Opbouw module

Doelgroep

Samenwerking

- Samenwerkingsopdrachten

- Expertmethode

- Werken met Scrum

Leerdoelen

- Inleiding

- Concepten

- Werkwijzen

- Vaardigheden

Veelgemaakte (denk)fouten

Hardwareplatform

- Keuze voor het platform

- Aanschaf van hardware: Micro:bit

- Aanschaf van hardware: Arduino

- Aanschaf van hardware: Lego Mindstorms

Uitgangspunten module

Verwachte voorkennis

Planning

Toetsing en beoordeling

Uitwerkingen

Terug naar voorpagina

Over dit lesmateriaal

## Inleiding

---

Physical Computing gaat over computersystemen die met de omgeving interacteren via sensoren en actuatoren. Het gaat om een breed terrein van mogelijke toepassingen: regel- en meetsystemen, (zorg)robots, zelfrijdende auto's, beveiligingssystemen, etc, etc. De toepassingen spreiden zich uit over zorg, veiligheid, transport, industrie, duurzaamheid, etc, etc. Bij het ontwikkelen hebben we op hoog niveau twee doelstellingen gehad:

1. De leerlingen meenemen in de mogelijkheden van physical computing in de maatschappij en de wereld om ons heen. We willen een tip van deze sluier oplichten en laten zien wat de huidige stand van zaken is als het gaat om physical computing. Wat kunnen 'we' allemaal, wat lukt nog niet? Neem bijvoorbeeld de zelfrijdende auto: de beloftes zijn hooggespannen, er zijn ook al diverse voorbeelden, tegelijkertijd zijn er nog uitdagingen, zoals het omgaan met de onzekerheden die rijden in het verkeer met zich meebrengen. Ander voorbeeld: chirurgen maken inmiddels gebruik van robots voor precisieoperaties. De nadruk ligt dus op de toepassingen en de mogelijkheden en beperkingen van physical computing. We hopen hiermee een brede groep leerlingen aan te spreken.
2. De leerlingen zelf een prototype te laten ontwerpen en ontwikkelen, op een gestructureerde manier. De beschikbaarheid van goedkope hardware en micro-controllers maakt het mogelijk om zelf nuttige en interessante systemen te bouwen. We willen leerlingen laten ervaren dat ze in staat zijn zelf iets te bouwen. Ook hier willen we een brede groep leerlingen aanspreken. We koppelen het aan een context zoals zelf-rijdende auto's, smart-cities, domotica, etc. Leerlingen kunnen kiezen wat voor systeem ze willen bouwen. Dat kan ook een kunst-project zijn. De leerlingen werken samen in multi-disciplinaire teams en kunnen zich daarbij toeleggen op de zaken die voor hen het meest motiverend zijn. Ze zijn in staat om eerst een model te beschrijven met behulp van toestandsdiagrammen en op basis daarvan een systeem te implementeren.

## Opbouw module

---

De module bestaat uit grofweg drie onderdelen:

- Inleiding op physical computing met verkenning van toepassingen in de maatschappij en in de wereld om ons heen
- Leren kennen van de bouwstenen voor physical computing en hoe een systeem te ontwerpen implementeren (vier cycli).
- Zelf een systeem ontwerpen en ontwikkelen (eindopdracht)

In elke cyclus benadrukken we een toepassingen en een specifieke sensor. De leerlingen bouwen in elke cyclus een eenvoudig systeem.

Zie meer over de opbouw en de vier cycli in het lesmateriaal: Inleiding Physical Computing -> [Over deze module](#).

## Doelgroep

---

Doelgroep is HAVO-5, VWO-5, VWO-6. We proberen een brede groep leerlingen aan te spreken door:

- te laten zien hoe physical in allerlei contexten/domeinen en de eigen leefwereld een rol speelt

(transport, veiligheid, kunst, zorg, domotica, etc).

- te benadrukken dat taken goed kunnen worden verdeeld:
  - sommige leerlingen geven de voorkeur aan het ontwerpen op basis van toestandsdiagrammen.
  - anderen zullen zich liever concentreren op het programmeren.
  - weer andere besteden het liefste aandacht aan de fysieke en mechanische kanten van een systeem

De module kan voldoende uitdaging bieden voor leerlingen die dat willen:

- op diverse plekken staan extra opdrachten beschreven
- de opdrachten zijn eenvoudig uit te breiden met extra eisen/wensen (eventueel door de leerlingen zelf bedacht)
- de eindopdracht biedt de mogelijkheid om eindeloos te verdiepen.

## Samenwerking

---

### Samenwerkingsopdrachten

De module bevat veel mogelijkheden voor leerlingen om samen te werken. Bij het ontwikkelen van een prototype is het goed mogelijk om taken te verdelen:

- maken van het toestandsdiagram
- aansluiten en testen van de sensoren
- het omzetten van een toestandsdiagram naar een programma
- fysiek bouwen van het prototype (krijgt in deze module geen expliciete aandacht)

Daarnaast is het vaak mogelijk om het ontwikkelen van een prototype op te delen in kleinere deelproblemen. In cyclus 3 gaat het bijvoorbeeld over een parkeersensor die bestaat uit een ultrasoonsensor en een piezobuzzer. Je kunt het probleem van de parkeersensor opdelen in twee deelproblemen:

1. Zorg dat het systeem detecteert dat een object binnen afstand\_A of afstand\_B is.
2. Zorg dat het systeem een zoemer aan en uit kan zetten, met twee verschillende frequenties (langzaam piepen en snel piepen).

Beide onderdelen kunnen apart worden ontwikkeld om vervolgens te worden samengevoegd.

In de module staan diverse samenwerkingsopdrachten beschreven:

- [Verken een toepassing van physical computing en bereid een presentatie voor](#)
- [Fantasie-apparaat](#)
- [Samenwerkingsopdracht: bouw een alarm](#) (specifiek Micro:bit)
- [Samenwerkingsopdracht: prototype lantaarnpaal maken](#) (specifiek Micro:bit)
- [Samenwerkingsopdracht: parkeersensor maken](#)

### Expertmethode

Veel van de opdrachten zijn goed in teams van 2 of 3 leerlingen te maken, waarbij de leerlingen de taken kunnen verdelen. Daarbij kan de expertmethode worden gebruikt. Leerlingen werken daarbij in groepjes van 2 of 3 leerlingen samen. Ieder van de leerlingen specialiseert zich in één onderdeel, samen met enkele leerlingen van andere groepjes. Als de experts voldoende zijn voorbereid keren ze

terug naar hun oorspronkelijke groepje. Ze brengen elkaar op de hoogte en maken vervolgens de gezamenlijke opdracht.

Een voorbeeld van de expertmethode is uitgewerkt in [Samenwerkingsopdracht: prototype lantaarnpaal maken](#).

## Werken met Scrum

Leerlingen werken samen in multi-disciplinaire teams. Belangrijk daarbij is dat ze transparant werken: het is helder wie wat doet en wat de voortgang daarbij is. Ze zijn als team aanspreekbaar. Ze werken efficiënt en effectief.

Binnen het bedrijfsleven wordt vaak Scrum ingezet. Ook binnen de klas kunnen de leerlingen met Scrum werken, sommige informatica-docenten passen dit met veel succes toe.

Ook deze module leent zich goed voor het werken met Scrum. Elke cyclus kan bijvoorbeeld binnen een sprint worden opgepakt. Op basis van 8 korte sprints doorlopen de leerlingen de module

Sprint 1	Inleiding + platform leren kennen
Sprint 2	Cyclus 1
Sprint 3	Cyclus 2
Sprint 4	Cyclus 3
Sprint 5	Cyclus 4
Sprint 6	Eindopdracht eerste versie
Sprint 7	Eindopdracht tweede versie
Sprint 8	Eindopdracht definitieve versie

Wellicht dat, als daar behoefte aan is, in het lesmateriaal voorbeelden worden opgenomen van product backlogs en sprint backlogs. Neem gerust contact met ons op via onderstaande mailadres.

**m.bruggink@tudelft.nl**

Doelgroep voor deze module zijn HAVO-5, VWO-5 en VWO-6. De module veronderstelt het kerndomein als voorkennis (zie [verwachte voorkennis](#)).

We willen graag een brede groep leerlingen aanspreken. Dat doen we door:

- te laten zien dat physical computing in allerlei contexten (transport, zorg, domotica, kunst, etc, etc) een rol speelt. Hopelijk spreekt leerlingen dit aan.
- te benadrukken dat taken kunnen worden verdeeld:
  - sommige leerlingen geven de voorkeur aan het ontwerpen via toestandsdiagrammen.
  - anderen maken liever een programma.
  - weer anderen werken liever aan de fysieke en mechanische onderdelen.

De module biedt voldoende de mogelijkheden tot differentiatie:

- in sommige hoofdstukken zijn extra opdrachten opgenomen

- veel van de opdrachten zijn eenvoudig met aanvullende eisen uit te breiden (eventueel zelf bedacht door de leerlingen)
- de eindopdracht biedt veel mogelijkheden tot uitdaging op het niveau dat past bij de specifieke leerlingen.

## Leerdoelen

### Inleiding

De leerdoelen zijn gebaseerd op de beschrijving die in het vernieuwde examenprogramma staan.

#### Eindtermen keuzedomein M: Physical Computing

##### *Sensoren en actuatoren*

De kandidaat kan sensoren en actuatoren waarmee een computersysteem de fysieke omgeving kan waarnemen en aansturen herkennen en functioneel beschrijven.

##### *Ontwikkeling Physical computing componenten*

De kandidaat kan fysieke systemen en processen modelleren met het oog op real time besturingsaspecten en kan met behulp van deze modellen, sensoren en actuatoren een computersysteem ontwikkelen om fysieke systemen en processen te bewaken en besturen.

Zie: <http://handreikingschoolexamen.slo.nl/informatica/Paginas/Keuzedomein-M.aspx>.

Specifiek zijn de leerdoelen

- De leerling kan diverse toepassingen van physical computing in diverse contexten / domeinen beschrijven.
- De leerling kan een prototype voor een fysiek systeem ontwerpen op basis van een toestandsdiagram.
- De leerling kan een prototype ontwikkelen en daarbij:
  - het probleem in kleinere problemen opdelen
  - in kleine stapjes werken / iteratief werken
  - gestructureerd werken
  - samenwerken in een multidisciplinair team
- De leerling kan de sensoren en actuatoren en de rol daarvan in een systeem benoemen.
- De leerling kan de kwaliteit van een (zelf ontworpen) systeem beschrijven op basis van de kwaliteit van de gebruikte sensoren en actuatoren, onder meer: functionaliteit, tijdigheid, nauwkeurigheid.

### Concepten

Overzicht van de belangrijkste concepten van physical computing.

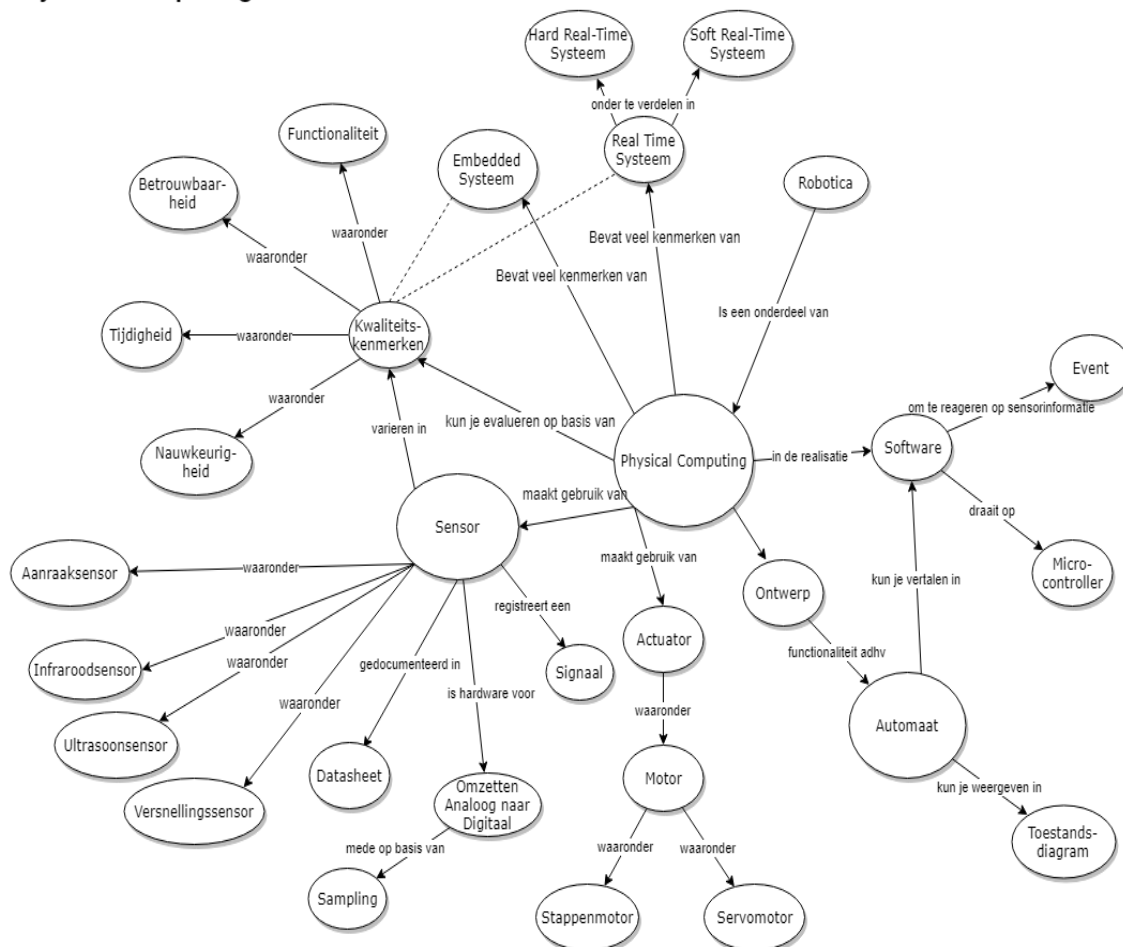
Concept	Beschrijving
Actuator	Hardware waarmee iets in de omgeving kan worden veranderd, bijvoorbeeld een motor, lampje, verwarmingselement, etc.
Analoog / Digitaal	Signalen in de fysieke wereld noemen we analoog, deze kunnen met behulp van sensoren omgezet worden in een digitaal signaal,

	bestaande uit enen en nullen.
Automaat / Eindigetoestandsautomaat	Een model van de functionaliteit van een systeem. Een automaat bestaat uit toestanden en toestandsovergangen die beschrijven wat de werking is van een systeem.
Datasheet	Een datasheet bevat informatie over (de kwaliteit van) een sensor, zoals de nauwkeurigheid, de omstandigheden waarin de sensor correct functioneert, de wijze van aansluiten van een sensor, etc.
Embedded systeem	Een computersysteem met een beperkte hoeveelheid functies, gemaakt voor een specifiek doel, zoals bijvoorbeeld de aansturing van een wasmachine.
Event (gebeurtenis)	Een gebeurtenis waar een systeem op kan reageren, bijvoorbeeld het registreren van een beweging bij een automatische deur, het bereiken van een bepaalde temperatuur in de kamer, etc.
Kwaliteitskenmerken, waaronder functionaliteit, robuustheid, betrouwbaarheid, tijdigheid, energieverbruik, nauwkeurigheid	Bepalen de waarde die een systeem binnen physical computing kan hebben voor een bepaalde toepassing.
Microcontroller	Een kleine computer die vaak wordt gebruikt voor systemen binnen physical computing waarop sensoren en actuatoren zijn aangesloten. Op de microcontroller draait software die de actuatoren aanstuurt op basis van de informatie die de sensoren leveren.
Motor (waaronder stappenmotor en servomotor)	Een actuator die een draaiende beweging tot stand brengt.
Ontwerp / ontwerpen	Beschrijving van (de werking van) een systeem. Bij physical computing wordt vaak gebruik gemaakt van een automaat als ontwerp voor de functionaliteit van een systeem.
Physical Computing	De tak van de informatica die zich bezig houdt met het (ontwikkelen van) computersystemen die via sensoren en actuatoren interactie hebben met de omgeving.
Real-time systeem	Een computersysteem waarbij de correcte werking niet alleen afhangt van de functionele werking van het systeem, maar ook van de snelheid waarmee dat gebeurt, zoals bijvoorbeeld het besturingssysteem van een zelfrijdende auto.
Robotica / Robot	De tak van informatica die zich bezig houdt met (het ontwikkelen van) systemen die kunnen zich zelf kunnen bewegen en kunnen reageren op waarnemingen in de omgeving. Robotica is in die zin een onderdeel van Physical Computing.
Sampling	Methode om analoge, continue signalen om te zetten in digitale, discrete informatie.

Sensor	Hardware waarmee een fysiek verschijnsel kan worden gemeten, bijvoorbeeld de temperatuur, een beweging, de afstand tot een object, etc
Signaal	Een stroom van informatie, bijvoorbeeld geluid, beeld, elektrische spanning, etc. Een computersysteem kan signalen meten met behulp van sensoren en op basis daarvan reageren.
Software	Computerprogrammatuur waarmee de gewenste werking van een systeem in detail wordt beschreven. Bij physical computing beschrijft de software hoe de actuatoren moeten handelen op basis van de gegevens die worden verkregen uit de software.
Timer	Een softwareonderdeel waarmee events kunnen worden gemaakt die aangeven dat een bepaalde hoeveel tijd is verstreken.
Toestandsdiagram	Een toestandsdiagram geeft een automaat visueel weer. Toestanden zijn bolletjes, toestandsovergangen zijn pijltjes.

Zie ook de concept map

## Concept Map Physical Computing





## Werkwijzen

Het werken met hardware als Arduino, en in minder mate Micro:bit en Lego Mindstorms is foutgevoelig. Het is daarom belangrijk om gestructureerd te werken. Het is echter niet eenvoudig om dat soort vaardigheden in lesmateriaal te vervatten. De docent kan de leerlingen hierin coachen. Het gaat onder meer om:

- Werken in kleine stapjes: bedenk wat de kleinste volgende stap is die je kunt nemen, werk dat uit en test dat. Ga pas verder als het goed werkt. Zorg dat je altijd terug kunt naar een vorige versie.
- Hierop aansluitend: test de sensoren en actuatoren afzonderlijk voordat je ze integreert.
- Ga uit van het toestandsdiagram voor het maken van het programma. Voorkom dat je veel tijd moet steken in het programmeren, leg de focus op het goed uitdenken van het toestandsdiagram.
- Werk netjes. Gebruik een breadboard voor het aansluiten van sensoren. Houdt het overzichtelijk.

### Over modelleren

Voordat de leerlingen een programma implementeren maken ze eerst een model van de werking van het systeem met behulp van toestandsdiagrammen. Aan de hand van het toestandsdiagram kunnen ze redeneren over de correctheid van het programma. Hierdoor kunnen ze fouten in vroeg stadium herkennen en verbeteren. Het omzetten van een toestandsdiagram naar een programma is relatief recht-toe-recht-aan. Door te werken met toestandsdiagrammen worden veel fouten voorkomen en zullen de leerlingen efficiënter werken. In praktijk zullen leerlingen vaak echter geneigd zijn om gelijk te gaan programmeren. De docent heeft een rol om de leerlingen te stimuleren eerst het model te maken alvorens te gaan programmeren.

Het maken van een toestandsdiagram is niet altijd makkelijk. Er is geen standaard route of algoritme voor het maken zo'n toestandsdiagram. Leerlingen vinden het lastig om te bepalen wat de toestanden zijn. Daarnaast maken ze soms toestandsdiagrammen die niet helemaal aan de regels voldoen. Die regels zijn op zich niet moeilijk, leerlingen zouden redelijk snel in staat moeten zijn om te bepalen of hun eigen toestandsdiagram voldoet aan de regels.

Op basis van een toestandsdiagram kun je redeneren over de werking van een systeem. Wat gebeurt er als ....? Dat kan bijvoorbeeld klassikaal, door een toestandsdiagram te laten zien (bijvoorbeeld een uitwerking van een leerling) en hierover een gesprek te voeren.

### Over ontwerpen

Idealiter krijgen de leerlingen de opdracht om een bepaald probleem op te lossen en daarbij zelf een oplossing te ontwerpen en een prototype te ontwikkelen. Het probleem is gegeven, de oplossing niet. Vaak echter wordt bij dit soort opdrachten (een deel van) de oplossing al gegeven. Bijvoorbeeld: bouw een auto die een lijn volgt op basis van lichtsensor. Beter zou zijn: bouw een voertuig dat automatisch naar een bepaalde plek kan rijden. Er zijn dan meerdere oplossingen mogelijk om dit probleem aan te pakken. Leerlingen kunnen deze oplossingen verkennen en vergelijken. In schoolpraktijk is dat echter vaak lastiger. De beschikbare materialen en hardware, de tijd en de expertise zijn beperkt. Het aantal haalbare oplossingsrichtingen wordt daardoor ook gelimiteerd. Daarnaast is het voor deze module belangrijk dat de oplossing wordt gezocht in de hoek van physical computing, terwijl het natuurlijk soms ook mogelijk is om oplossingen zonder physical computing te realiseren. Kijk bijvoorbeeld naar de NLT-module Technisch Ontwerpen in de Biomedische Wetenschap, waarbij leerlingen een oplossing moeten vinden voor mensen met een beperking. Hoe kan iemand die niet kan zien

voorkomen dat het kopje overloopt bij het inschenken van de koffie? Daar zijn verschillende oplossingen voor te bedenken, zowel met als zonder physical computing.

Een ander aspect is dat leerlingen geneigd zijn om voor de eerste-de-beste oplossing te gaan zonder eerst de mogelijkheden te verkennen. Ze moeten echter leren meerdere oplossingsrichtingen te onderzoeken voordat ze een keuze maken. Dit divergeren is een belangrijke vaardigheid.

Belangrijk in het ontwerpproces is dat ze hun keuze onderbouwen. Waarom wordt voor de ene oplossingsrichting gekozen ten nadele van een andere oplossingsrichting? Specifiek voor deze module is de keuze voor bepaalde sensoren en actuatoren. Waarom wordt voor het ene type sensor gekozen? Dan is kennis over de eigenschappen van de sensor belangrijk.

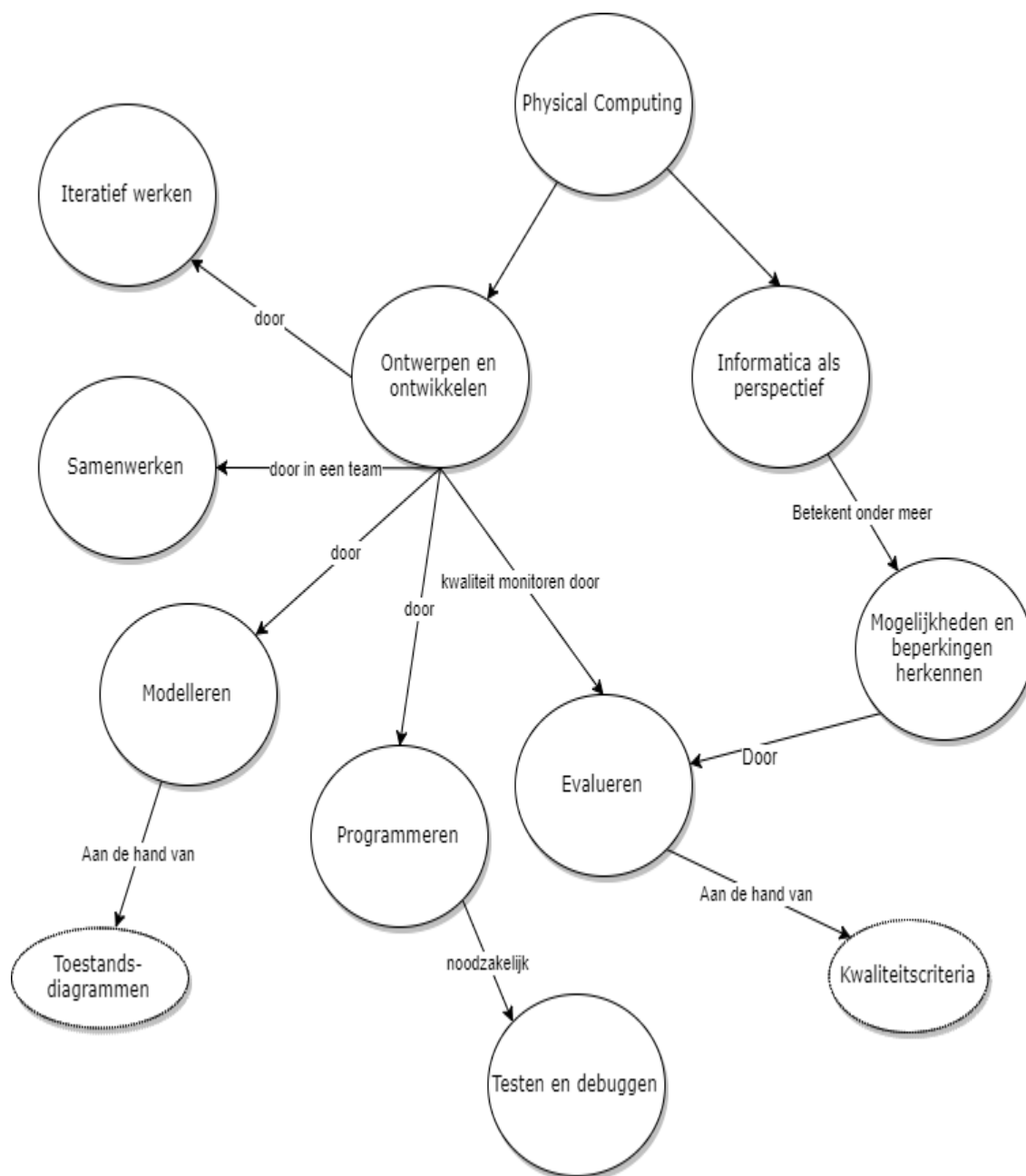
Overigens hebben leerlingen vaak geen goed beeld van wat de mogelijkheden zijn. Daarom laten we de leerlingen veel voorbeelden zien van systemen/prototypen die ze in principe zelf ook kunnen maken.

## Vaardigheden

In het vernieuwde examenprogramma worden drie vaardigheden uitgelicht, die allen ook een plek hebben in deze module:

1. Ontwerpen en ontwikkelen: de leerlingen gaan zelf een prototype ontwerpen en ontwikkelen
2. Informatica hanteren als perspectief: de leerlingen leren wat de mogelijkheden en beperkingen zijn van physical computing
3. Samenwerken en interdisciplinariteit: de leerlingen werken in teams bij het ontwerpen en ontwikkelen van het prototype

Dat is ook te zien in het volgende overzicht:



## Veelgemaakte (denk)fouten

Bij het testen zijn we enkele denkfouten bij leerlingen tegengekomen die wellicht vaker zullen voorkomen. Het is goed om daar als docent bewust van te zijn en op in te kunnen spelen.

1. Een schakeling zien als een natuurkundige schakeling. Neem bijvoorbeeld een hotelschakeling in huis. Daar komt geen microcontroller bij aan te pas. Bij physical computing gaan we uit van een microcontroller die het gedrag van het systeem bepaalt op basis van een programma. Dat betekent dat alle sensoren en actuatoren gekoppeld zijn aan de microcontroller en niet direct aan elkaar zoals bij een natuurkunde schakeling.
2. De leerlingen hebben soms geen goed beeld van wat voor sensoren er zijn. Vraag bijvoorbeeld naar sensoren voor een grasmaaierrobot, en wellicht wordt dan een sensor genoemd die de hoogte van het gras kan meten. Op zich zou het mogelijk moeten zijn om iets te ontwikkelen waarmee de hoogte van het gras wordt gemeten, dit zal dan waarschijnlijk uit een combinatie van diverse sensoren en software zijn. Het gaat dus om het abstractieniveau.

3. Bij het maken van toestandsdiagrammen zijn we verschillende soorten fouten tegengekomen. Het maken van een toestandsdiagram is overigens niet eenvoudig, het vergt de nodige oefening.

- Het toestandsdiagram wordt gezien als een soort stroomdiagram (flow charts), dat sequentieel werkt. Belangrijk bij het werken met toestandsdiagrammen is na te denken over de toestand waarin een systeem kan verkeren. Toestandsdiagrammen en stroomdiagrammen zijn wezenlijk anders in de manier waarop ze moeten worden geïnterpreteerd.
- Objecten (zoals een afstandsbediening) worden soms gemodelleerd als een toestand
- Signalen tussen twee onderdelen (bijvoorbeeld een afstandsbediening en een tv) worden gemodelleerd als een toestandsovergang.
- Soms zijn leerlingen terughoudend in het zelf toevoegen van extra toestanden, terwijl dat wel nodig is om het systeem goed te modelleren.
- Soms worden toestandsovergangen gemaakt zonder specifieke gebeurtenis, de overgang verloopt als het ware automatisch. Een toestandsovergang moet altijd gekoppeld zijn aan een gebeurtenis.
- Niet alle gebeurtenissen worden gekoppeld aan alle toestanden (zie [Maak een tabel met alle mogelijke toestandsovergangen](#)).
- De begintoestand wordt gauw vergeten.

## Hardwareplatform

---

### Keuze voor het platform

We werken op basis van drie mogelijke platforms:

Micro:bit	De Micro:bit is een goedkope microcontroller met enkele sensoren geïntegreerd op het bordje. Daardoor is de Micro:bit minder foutgevoelig dan de Arduino. Samen met de visuele omgeving maakt dat de Micro:bit toegankelijk, het instapniveau is laag. Het is mogelijk om externe sensoren en actuators aan te sluiten. Inmiddels zijn al behoorlijk wat voorbeelden en handleidingen te vinden online.
Arduino	De Arduino is een goedkope microcontroller waarvoor veel sensoren en actuators verkrijgbaar zijn. Er zijn veel voorbeelden en handleidingen te vinden op internet. Het programmeren en (fysiek) aansluiten van sensoren en actuators is relatief lastig en foutgevoelig.
Legomindstorms	Onderdelen in de Lego Mindstorms zijn makkelijk aan te sluiten. Het fysiek bouwen van een systeem is relatief eenvoudig. Het programmeren kan in een visuele omgeving, maar ook in een tekstuele omgeving. Nadeel van deze hardware is dat het relatief kostbaar is.

Voor alledrie platforms is het lesmateriaal min of meer opdezelfde manier uitgewerkt. De basisconcepten zijn uitgewerkt in platformafhankelijke hoofdstukken. Het is aan de docent en eventueel de leerlingen om een keuze te maken voor één platform.

Het is belangrijk om te realiseren dat leerlingen bij het ontwikkelen van een systeem veel problemen kunnen tegenkomen. Dat maakt het ontwikkelen van een systeem met sensoren en actuators relatief lastig. Als het systeem niet werkt kan de bron in de hardware liggen, in het verkeerd aansluiten van sensoren of actuators of in het verkeerd programmeren. Aan de andere kant kan daar ook juist weer de uitdaging liggen.

## Aanschaf van hardware: Micro:bit

Er zijn twee mogelijkheden. De voorkeur gaat naar optie 1, die is echter iets duurder.

### Optie 1: Micro:Bit inclusief uitvindingskit van Kitronik

Aangeraden wordt om bij elke Micro:Bit een uitvindingskit te kopen. Deze kit bevat namelijk een breadboard en breakoutboard, waardoor het aansluiten van externe sensoren en actuatoren een stuk makkelijker wordt. De uitvindingskit bevat daarnaast een aantal sensoren, actuatoren, ledjes, weerstandjes, draadjes etc.

Product	Kosten in euro's			
	1	10	15	20
BBC Micro:Bit Go Bundel*	22.95	189.95	304.70**	379.90
Uitvindings Kit voor de BBC micro:bit	29.95	284.47	426.71	568.94
PIR Bewegingssensor	6.95	62.56	93.84	125.11
Ultrasoon Sensor – HC-SR04	4.95	47.07	70.60	94.14
Jumperwires – 10 stuks – F/F – 15cm	2.95	26.50	39.75	53.00
Jumperwires – 10 stuks – M/F – 15cm	2.95	26.50	39.75	53.00
Jumperwires – 10 stuks – M/M – 15cm	2.95	26.50	39.75	53.00
3x AAA batterijhouder	2.50	23.11	34.67	46.22
Micro USB kabel voor aansluiten Micro:bit op computer	2.95	29.50	44.25	59.00
<b>Totaal</b>	<b>79.10</b>	<b>716.16</b>	<b>1094,00</b>	<b>1432.30</b>

\*De BBC Micro:Bit 10-pack is hetzelfde als 10x de Micro:Bit Go Bundle. De 10-pack is een stuk goedkoper dan wanneer de Micro:Bit los gekocht wordt.

\*\*Een 10-pack + 5 BBC Micro:Bit Go Bundles

In de uitvindings kit zit sowieso 10 M/F en 10 M/M draadjes. Echter de draadjes kunnen makkelijk zoekraken of kapot gaan (als er niet goed mee wordt omgegaan), dan is het verstandig om extra draadjes te bestellen. Ook F/F draadjes zijn handig om te hebben.

### Optie 2: Micro:Bit exclusief uitvindingskit van Kitronik

Het is ook mogelijk om zonder de uitvindingskit te werken. Dan zijn er wel kabeltjes met krokodillebekjes nodig om de externe sensoren te kunnen aansluiten. Daarnaast heb je dan geen beschikking over een piezobuzzer die in cyclus 3 wordt gebruikt. Alternatief is om een headsetje / oortjes te gebruiken die de meeste leerlingen wel hebben.

Product	Kosten in euro's			
	1	10	15	20
BBC Micro:Bit Go Bundel*	22.95	189.95	304.70**	379.90
PIR Bewegingssensor	6.95	62.56	93.84	125.11
Ultrasoon Sensor – HC-SR04	4.95	47.07	70.60	94.14
Kabeltjes met krokodillenklemmetjes (12 stuks)	4.95	44.50	66.75	89.00
Krokodillenklemmetje naar Female Jumperwire (10 stukes)	4.95	44.50	66.75	89.00
3x AAA batterijhouder	2.50	23.11	34.67	46.22

Micro USB kabel voor aansluiten Micro:bit op computer	2.95	29.50	44.25	59.00
<b>Totaal</b>	<b>50.20</b>	<b>441.20</b>	<b>681.60</b>	<b>882.40</b>

\*De BBC Micro:Bit 10-pack is hetzelfde als 10x de Micro:Bit Go Bundle. De 10-pack is een stuk goedkoper dan wanneer de Micro:Bit los gekocht wordt.

\*\*Een 10-pack + 5 BBC Micro:Bit Go Bundles

### Overige opmerkingen

Ieder groepje leerlingen heeft (minstens) één setje nodig.

De PIRsensor (cyclus 2) en Ultrasoonsensor (cyclus 3) zijn best prijzig. Als leerlingen deze sensoren enigszins kunnen delen, dan is het niet nodig om voor elke Micro:Bit een eigen sensor aan te schaffen.

Besteld via: <https://www.kiwi-electronics.nl/>.

## Aanschaf van hardware: Arduino

Per leerling of per groepje leerlingen.

### Basis:

- Arduino Uno, Genuino Uno of een vergelijkbare microcontroller
- Breadboard
- Aansluitdraadjes in verschillende kleuren, varianten (male-male, female-male, female-female) en lengten
- Een set ledjes
- Een set weerstanden van 220  $\Omega$ , 330  $\Omega$  of 470  $\Omega$

### Sensoren

Cyclus 1:

- 2x Aanraaksensor (Capacitive Touch Switch Sensor) **of** 2x drukknop

Cyclus 2:

- 1x Passieve Infraroodsensor (PIR-sensor)
- Eventueel: een actieve infraroodsensor

Cyclus 3:

- 1x ultrasone afstandssensor
- Eventueel: een piezobuzzer

Cyclus 4:

- 1x versnellingsensor/gyroscop is de MPU6050. Die wordt meestal geleverd op een zogenaamd breakout board en dat board is de GY-521

Verder:

- Eventueel één of meerdere motortjes
- Eventueel ook een 16x2 LCD display met I2C aansluiting maar dit is hier slechts als voorbeeld gebruikt.

## Aanschaf van hardware: Lego Mindstorms

Lego Mindstorms EV3 Basisset – 45544

Deze (educatieve) set bevat 2 touch-sensoren, 1 kleur-sensor, 1 ultrasoon-sensor en 1 gyrosensor. Met deze set is een eenvoudige, maar doeltreffende robot te bouwen. Een IR-sensor ontbreekt. Het is nodig om een oplader voor de accu te bestellen, deze wordt niet met de set meegeleverd. De set is te bestellen bij diverse bedrijven, o.a. De vermelde prijzen zijn inclusief BTW. Het vergelijken van de prijzen is aanbevolen.

- Eurofysica, Heutink € 459,80
- CMA prijs opvragen

Lego Mindstorms EV3 transformator – 8887

- Eurofysica, Heutink € 38,70

Lego Mindstorms EV3 IR-sensor – 45509

- Eurofysica € 41,75
- Heutink € 44,76

Via websites, zoals van [lego-discounter.com](http://lego-discounter.com) of [misterbricks.nl](http://misterbricks.nl), zijn de set en de losse onderdelen soms een stuk voordeliger aan te schaffen.

HiTechnic (<https://modernroboticsinc.com/hitechnic-sensors>) levert o.a. infraroodsensoren voor de EV3, zoals

- HiTechnic NXT PIR Sensor (eenvoudige passieve sensor, ook geschikt voor de EV3)
- HiTechnic NXT IRSeeker V2

Via [lego-discounter.com](http://lego-discounter.com) en [robotdiscounter.com](http://robotdiscounter.com) zijn een aantal van deze HiTechnic sensoren leverbaar.

## Uitgangspunten module

---

Bij het ontwikkelen van de module hebben we de volgende uitgangspunten gehanteerd:

- We willen laten zien wat toepassingen zijn van physical computing in de wereld om ons heen. Denk aan toepassingen in de zorg, transport, medische wereld, veiligheid, etc. Daarmee willen we een brede groep leerlingen aanspreken.
- We laten zien dat het voor de leerlingen heel goed haalbaar is om zelf zo'n systeem te ontwikkelen aan de hand van diverse voorbeelden (waaronder ook kunstzinnige/creatieve systemen).
- We willen dat de leerlingen zelf een systeem of prototype ontwerpen en ontwikkelen. Daarbij zijn werkwijzen als iteratief werken, samenwerken en evalueren van belang. Het is daarbij ook mogelijk dat de klas een samenhangend prototype ontwerpt en ontwikkelt, bijvoorbeeld rondom het thema smart-city.
- We willen dat de leerling in staat is om keuzes bij de ontwikkeling te onderbouwen, specifiek de keuzes voor de sensoren en actuatoren.
- We willen dat de leerling in staat is om een systeem te evalueren op aspecten als tijdigheid, nauwkeurigheid en functionaliteit.
- De basis wordt zo veel mogelijk platform-onafhankelijk behandeld. Het platform-afhankelijke

deel wordt voor drie platforms uitgewerkt: Arduino (hogere moeilijkheidsgraad), Lego-mindstorms en Micro:Bit (lagere moeilijkheidsgraad).

- We bieden de bouwstenen aan op basis waarvan de leerling een systeem of prototype kan ontwerpen en ontwikkelen.
- We willen dat de leerling in staat is om een systeem te modelleren/beschrijven met behulp van toestandsdiagrammen (eindige automaten).
- De module gaat slechts beperkt in op de theorie van real-time systems en embedded systems. We hebben gekozen voor een praktische insteek. Centrale begrippen zijn: physical computing, sensoren, actuatoren, tijdigheid, automaten/toestandsdiagrammen.
- Belangrijk is dat leerlingen op basis van kennis van sensoren en actuatoren een afweging kan maken bij de selectie hiervan in de ontwikkeling van het systeem.
- We maken waar het kan gebruik van bestaande bronnen. Er is al veel over dit onderwerp geschreven, daar willen we graag gebruik van maken.

## Verwachte voorkennis

---

De stof uit het kernprogramma wordt als voorkennis verondersteld. Hieronder staat een globaal overzicht van de relaties met het kernprogramma.

- Domein B (Grondslagen): Specifiek de stof uit B3: Eindige Automaten is belangrijk. In deze module moeten de leerlingen zelf een toestandsdiagram kunnen maken en de correctheid ervan kunnen bepalen. Op die manier kunnen ze een algoritme (B1) ontwerpen.
- Domein C (Informatie): Specifiek is gegevensrepresentatie belangrijk: hoe worden gegevens verkregen uit sensoren gerepresenteerd.
- Domein D (Programmeren): Leerlingen moeten een toestandsdiagram kunnen vertalen naar een programma. De basiselementen van programmeren worden daarbij als voorkennis beschouwd, met name if-then-else structuren, variabelen en logische operatoren (EN, OF). Het debuggen krijgt bij physical computing relatief veel aandacht vanwege de foutgevoeligheid.
- Domein E (Architectuur): Specifiek de hardwarelaag in de vorm van sensoren en actuatoren, de vertaling tussen digitaal en analoog, en de rol van de micro-controller zijn van belang.
- Domein F (Interactie): Specifiek de interface tussen gebruiker en systeem als ook de maatschappelijke aspecten zijn van toepassing voor deze module.

## Planning

---

Totaal gaan we uit van 60 studielastuur (SLU) voor deze module. Het is echter geen module die de leerlingen per se van a tot z moeten doorlopen. Er zijn diverse mogelijkheden voor de docent en leerlingen om nadruk te leggen op bepaalde aspecten en andere delen over te slaan. Dit geldt zeker voor de eindopdracht, die kun je zo uitdagend maken als je wilt.

Cyclus 1 en 2 moeten in ieder geval worden doorlopen om te kunnen voldoen aan [de doelen zoals beschreven in het examenprogramma domein M](#). Cyclus 3 biedt wat meer verdieping, cyclus 4 wat meer verbreding. Beide kunnen onafhankelijk van elkaar worden doorlopen (nadat cyclus 1 en 2 zijn doorlopen).

Vwo-leerlingen hebben 440 SLU voor het vak informatica, Havo-leerlingen 320 SLU. Daarvan is 200 SLU voor het kernprogramma. Vwo-leerlingen hebben ruimte voor 4 keuze-modules, havo-leerlingen doen 2 keuze-modules. Ter vergelijking, een NLT-module heeft de omvang van 40 SLU. In praktijk

---



wordt de helft daarvan besteed aan lesuren. Voor een 60 SLU module zijn dus zo'n 30 lesuren beschikbaar.

De planning hangt echter ook af van de keuze van de hardware. Het werken met Arduino kost over het algemeen meer tijd dan het werken Micro:bit en Lego-mindstorms.

Onderstaande overzicht is een mogelijke planning voor het werken met de Micro:bit.

Inleiding (inclusief platform leren kennen)	+/- 15 SLU
Cyclus 1	+/- 8 SLU
Cyclus 2	+/- 8 SLU
Cyclus 3	+/- 7 SLU
Cyclus 4	+/- 7 SLU
Eindopdracht	+/- 15 SLU
<b>Totaal</b>	<b>60 SLU</b>

Er zijn ook andere varianten mogelijk. Daarbij hoeft niet altijd 60 SLU worden besteed, minder (of meer) kan ook. In plaats van een open eindopdracht kan de docent er ook voor kiezen een specifieke opdracht te geven, waardoor dit minder tijd kost. De docent kan dan precies omschrijven wat het op te leveren systeem moet doen en welke elementen het bevat, waardoor het ontwikkelen van het systeem minder tijd kost. Ook is het mogelijk om de eindopdracht te combineren met een opdracht uit een andere module, zoals bijvoorbeeld Internet of Things.

## Toetsing en beoordeling

---

De eindbeoordeling bestaat voorals nog alleen uit [een eindopdracht en een model voor het beoordelen van het ontwerpen en ontwikkelen van een prototype](#). De beoordeling is gebaseerd op de uitvoering en resultaten van de eindopdracht. Wellicht volgt later nog een meer theoretische toets.

## Uitwerkingen

---

De uitwerkingen van de opgaven zijn voor docenten op verzoek verkrijgbaar via onderstaande mailadres.

**m.bruggink@tudelft.nl**

## Terug naar voorpagina

---

[Ga terug naar de voorpagina](#) met verwijzingen naar alle hoofdstukken voor deze module.

## Over dit lesmateriaal

---

### Colofon

<b>Auteur</b>	Physical Computing
<b>Laatst gewijzigd</b>	17 July 2019 om 16:08
<b>Licentie</b>	Dit lesmateriaal is gepubliceerd onder de Creative Commons Naamsvermelding 4.0 Internationale licentie. Dit houdt in dat je onder de voorwaarde van naamsvermelding vrij bent om: <ul style="list-style-type: none"><li>• het werk te delen - te kopiëren, te verspreiden en door te geven via elk medium of bestandsformaat</li><li>• het werk te bewerken - te remixen, te veranderen en afgeleide werken te maken</li><li>• voor alle doeleinden, inclusief commerciële doeleinden.</li></ul>

[Meer informatie over de CC Naamsvermelding 4.0 Internationale licentie](#)

### Aanvullende informatie over dit lesmateriaal

Van dit lesmateriaal is de volgende aanvullende informatie beschikbaar:

<b>Eindgebruiker</b>	leerling/student
<b>Moeilijkheidsgraad</b>	gemiddeld
<b>Studiebelasting</b>	4 uur en 0 minuten