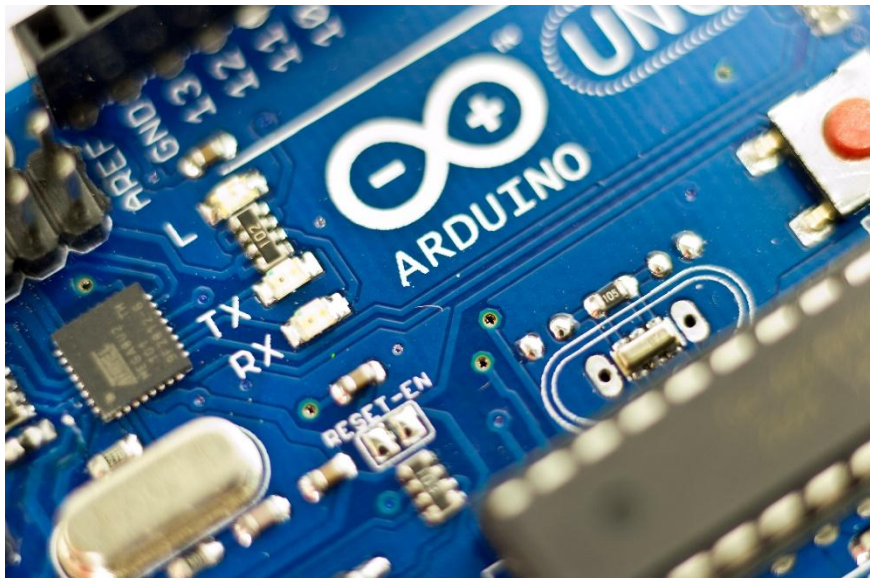


PO INTRO PHYSICAL COMPUTING MET ARDUINO

v1.5 - 2026



Inhoud

Introductie	1
Opzet	2
Basisopdrachten	2
Eindopdracht(je)	4
Beoordeling	5
Deadline en inleveren:.....	5

Introductie

We gebruiken [de Arduino module uit onze online lesmethode Fundament](#) (Domein A, hoofdstuk 5: Arduino) als introductie tot Arduino. De lesstof gaat uit van een echte fysieke Arduino, maar wij werken (grotendeels) met een [simulatie in Tinkercad](#). De verschillen zijn niet groot, maar wel handig om wat uitleg erbij te hebben. [Wt heeft een YouTube filmpje gemaakt met uitleg en de uitwerking van een deel van de eerste opdracht.](#)

Voordat je begint. [Meld je even aan voor de juiste "SGDB Arduinoklas" op Tinkercad](#). Dan kan je docent met je meekijken (handig als je een vraag hebt) en is alles automatisch ingeleverd voor de beoordeling.

Opzet

Deze PO bestaat uit:



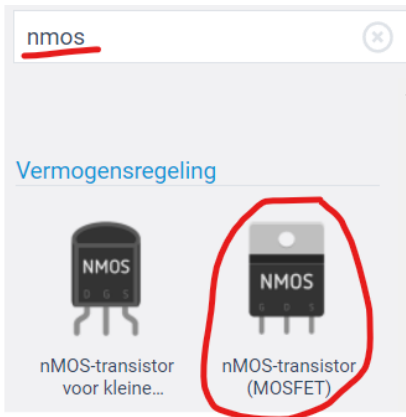

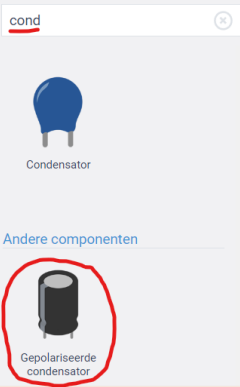
- 8 basisopdrachten, deze worden volledig uitgelegd in de lesmethode en hoeft je slechts na te bouwen in Tinkercad. Bij elke opdracht zit op het einde ook nog een uitbreidingsopdracht. Hierbij moet je toepassen wat je hebt geleerd en daarbij zelf de code en schakeling maken/uitbreiden. Je krijgt een deel van de punten voor de basisopdracht en een deel voor de uitbreiding (zie details en tips per opdracht onder "[beoordeling](#)")
- Een kleine eindopdracht. Hierbij heb je de keuze uit vier opdrachten (3.1 t/m 3.4) als uitgangspunt. Je bouwt de opdracht na en breidt hem zelf verder uit.

Er is [een studiewijzer beschikbaar](#) om je te helpen plannen.

Basisopdrachten

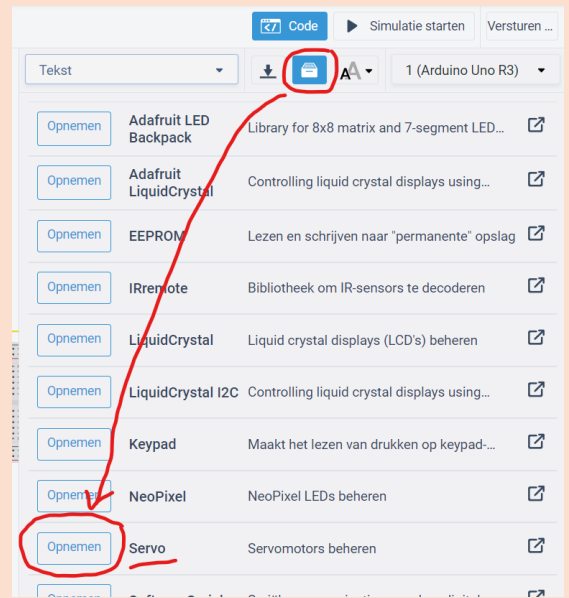
In de tabel hieronder vind je tips voor elk van de basisopdrachten. Bekijk steeds de tips van een opdracht voordat je hem maakt.

Onderdeel/opgave	Opmerkingen, tips en hints		
Paragraaf 1.1 en 1.2 (lezen)	Even doorlezen, zodat je de basics van de Arduino begrijpt. Paragraaf 1.3 en 1.4 mag je voor nu overslaan. Die gaan over het aansluiten met een fysieke Arduino, terwijl wij met een gesimuleerde aan de slag gaan. Lees deze later als je met een fysieke Arduino gaat werken nog eens door		
Paragraaf 2.1 (opdracht)	Maak deze opdracht inclusief de uitbreidingen onderaan. Je bewaart de eindversie met het stoplicht om in te leveren. Tussenversies hoeft je niet te bewaren. In het uitlegfilmpje van Wt wordt het eerste deel van de opdracht voorgemaakt en meteen Tinkercad uitgelegd. Prima startplek voor deze opdracht.		
Paragraaf 2.2 (opdracht)	Maak de schakeling uit deze opdracht, inclusief de uitbreidingen op het einde. Bewaar de versie met de uitbreidingen (met de variabelen en extra led)		
Paragraaf 2.3 (opdracht)	Maak de schakeling inclusief de uitbreidingen (met variabele en for-loop die de led ook weer dimt) <i>Hint:</i> De for-loop in C++ werkt hetzelfde als de for loop uit Python met een iets andere notatie. De volgende stukjes code doen hetzelfde (een variabele langzaam van 0 tot 255 laten lopen en voor elke stap de code in het blok uitvoeren: <table border="1"><tr><td><pre>// C++ for-loop for(int waarde=0; waarde<255; waarde=waarde + 1) { // doe iets }</pre></td><td><pre># Python for-loop: for waarde in range(0,255): #doe iets</pre></td></tr></table>	<pre>// C++ for-loop for(int waarde=0; waarde<255; waarde=waarde + 1) { // doe iets }</pre>	<pre># Python for-loop: for waarde in range(0,255): #doe iets</pre>
<pre>// C++ for-loop for(int waarde=0; waarde<255; waarde=waarde + 1) { // doe iets }</pre>	<pre># Python for-loop: for waarde in range(0,255): #doe iets</pre>		
Paragraaf 2.4 (opdracht)	Maak de schakeling met de potmeter. Maak ook de uitbreiding met de instelbare delay. Let op de potmeter op het voorbeeldplaatje in Fundament is net wat anders dan die uit Tinkercad. Bij de Fundament versie zit het middelste pootje rechts en moet je het draadje ervoor dus rechts aansluiten. In Tinkercad zitten alle pootjes aan dezelfde kant en moet je het draadje ervoor links aansluiten.		

	<p>Zie afbeeldingen hieronder. Rechts is de Tinkercad versie:</p> 
<p>Paragraaf 2.5 (opdracht)</p>	<p>Maak de schakeling inclusief de uitbreiding die bij de drempelwaarde het ledje aan laat gaan. Omdat we in tinkercad natuurlijk geen echte lichtwaarden hebt, mag je zelf een drempelwaarde kiezen voor de led. De LDR heet in Tinkercad “Lichtgevoelige weerstand”:</p> 
<p>Paragraaf 2.6 (opdracht)</p>	<p>Net als bij 2.5 moet je hier zelf even een drempelwaarde kiezen voor het aanzetten van de LED, omdat we met een gesimuleerde arduino werken.</p>
<p>Paragraaf 2.7 (opdracht)</p>	<p>Maak de schakeling met de motor, inclusief de uitbreiding met 2 leds. Let op, voor de juiste transistor moet je in Tinkercad even zoeken. Gebruik de zoekbalk en zoek naar “nmos”. Kies dan de transistor waar “MOSFET” bij staat:</p> 
<p>Paragraaf 2.8 (opdracht)</p>	<p>Maak de schakeling inclusief de uitbreiding met de potmeter. Maak een losse versie van de uitbreiding met het gas- en rempedaal.</p>   <p>Spiek bij 2.4 hoe de potmeter ook alweer werkte. Tip bij het gas- en rempedaal: Het is handig om een variabele “snelheid” te maken die de “snelheid” van de auto (en dus de stand van de servo) bijhoudt. Gas geven verhoogt de variabele, remmen verlaagt hem, geen enkele knop indrukken verlaagt hem langzaam. Je schrijft continue de waarde hiervan naar de servo.</p> <p>Paar dingen om op te letten bij Tinkercad:</p> <ul style="list-style-type: none"> - Gebruik als servo de “Micro Servo” uit het basispalet - Gebruik als condensator de “gepolariseerde condensator” (gebruik de zoekfunctie) - Om in Tinkercad de Servo library te importeren moet je het volgende doen: Open het codevenster, klik op het “Bibliotheken” knopje bovenaan. Zoek “Servo” in de lijst en

klik "opnemen":

Hint voor de uitbreiding:
De potmeter geeft een waarde tussen de 0 en de 1024 en de servo verwacht een waarde tussen de 0 en 180. Om deze om te rekenen kun je doen:
 $\text{potWaarde} / 1024 * 180$
Dit werkt alleen als je *potWaarde* als een *float* declareert ipv een *int* (zoals in opg 2.4 gebeurt).



Eindopdracht(jes)

1. Maak een keuze uit 1 van de opdrachten uit Hoofdstuk 3 (3.1, 3.2, 3.3. of 3.4) en maak deze.
 - Bouw de schakeling van de door jou gekozen opdracht. Hier staat wat minder uitleg bij dan bij de basisopdrachten. Je zult zelf dus even moeten puzzelen.
 - Breid het systeem uit met een zelfbedachte en geïmplementeerde uitbreiding
2. Als je nog genoeg tijd hebt nadat je bovenstaande uitbreiding hebt gedaan: verzin zelf een interessante schakeling en bouw deze. Geef een toelichting in je verslag.

Beoordeling

Basisopdrachten: Max 5,5 punten:

Opdracht	Maximale punten voor alleen basisversie:	Maximale punten voor basisversie + uitbreiding:
2.1	0,3	0,6
2.2	0,3	0,7
2.3	0,3	0,7
2.4	0,3	0,7
2.5	0,3	0,7
2.6	0,3	0,7
2.7	0,3	0,7
2.8	0,3	0,7

Eindopdracht: Max 4,5 punten:

Onderdeel:	Maximale punten:
Correcte uitwerking van een van de 4 keuzeopdrachten uit H3 in TinkerCad	1,5
Interessante/nuttige eigen uitbreiding op de gekozen eindopdracht (eventueel een leuk eigen project nadat je de uitbreiding klaar hebt)	3,0

Deadline en inleveren

Deadline: Wo 17 juni 23:59

(Net voor de toetsweek. Er is geen informaticoets in de TW, dus zie deze PO als eindtoets)

- Inleveren van de Tinkercad opdrachten gaat automatisch, als je je maar hebt aangemeld voor het juiste "[SGDB Tinkercad klasje](#)".
- Verder lever je een kort verslagje in in de Magister ELO met een korte toelichting op je eigen uitbreiding(en) op de eindopdracht en je eventuele zelfgebouwde extra opdracht