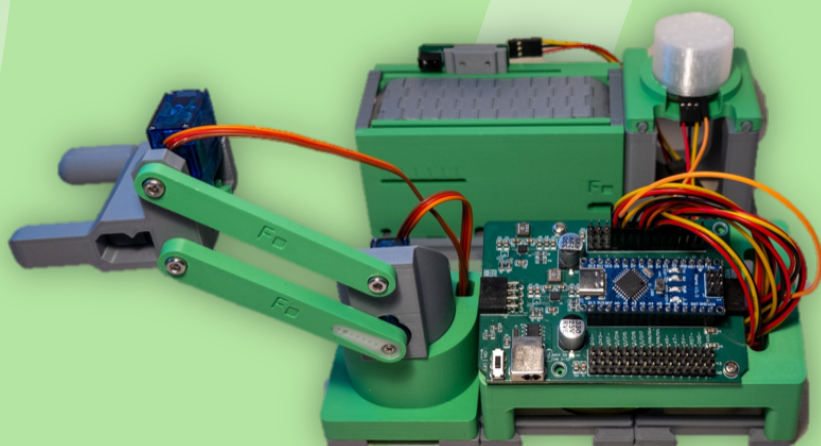


LEAPHY FACTORY



Hallo, mijn naam is **Bas** en welkom bij de **Leaphy Factory!**

Ik kom zelf van het **VMBO-TL** en ben daarna direct de techniek in gegaan, omdat ik dat supergaaf vond. Ik heb **Technische Engineering (mbo-niveau 4)** gedaan en ben daarna doorgestoomd naar de hogeschool (**HBO Electrical Engineering**). Om het af te ronden heb ik daar nog een **master** bovenop gedaan. Ik heb heel veel verschillende scholen gezien en daar veel van geleerd.

Om de cirkel rond te maken, werk ik nu als **technisch ontwerp docent op het MBO**. Vanuit mijn passie voor de kleine mechanica wilde ik een toegankelijke en modulaire fabriek maken die voor iedereen bereikbaar is. Wat we proberen is om echt dicht bij de **realiteit** te werken: de logische stappen te volgen en te ontdekken hoe complex een systeem wel niet is.

Heel veel bouw- en programmeerplezier toegewenst!

*Robotontwerp en lesmateriaal: Bas de Graaf
Eerste druk januari 2025
Stichting Leaphy*

Executive Summary

De **Leaphy Factory** is een modulair educatief roboticasysteem dat een industriële productielijn simuleert. Het systeem is ontworpen om leerlingen praktische ervaring te bieden met mechanica, elektronica en programmeren.

Het systeem bestaat uit vier hoofdmodules die met elkaar samenwerken:

- **Main Module:** De centrale besturingseenheid, gebaseerd op een Arduino Nano en het 'Leaphy Murphy shield'.
- **3-DOF Manipulator:** Een robotarm aangedreven door servomotoren.
- **Conveyor Belt:** Een lopende band voor transport.
- **Andon Light:** Een signaaltoren.

De handleiding bevat gedetailleerde bedradingsschema's om de servo's en sensoren correct aan te sluiten op de controller. Voor de aansturing wordt gebruikgemaakt van **Leaphy Easybloqs**, een visuele programmeeromgeving waarmee de hardware laagdrempelig geprogrammeerd kan worden.

Table of contents

Construction pages	4
<i>In the box</i>	6
<i>Connection overview</i>	8
A deep dive into Leaphy	10
The Modular Base	12
The Control Hub	14
Andon Light	16
<i>Sub-programs</i>	18
The Conveyor Belt	20
<i>Variables</i>	22
<i>Read Sensor</i>	22
<i>Output on screen</i>	22
<i>IF Else</i>	24
The 3-DOF Manipulator	26
Sine Wave Control	34
<i>Range and equilibrium (A & D)</i>	36
<i>Period and the Frequency (B)</i>	40
<i>The Phase Shift (C)</i>	42
Now it's up to you	46
Software Explanation	48
Complete Software	51
Upgrades	52

Construction pages

Hey toekomstige tech-held!

Je hebt een **fysiek werkboek** voor alle theorie, opdrachten en leerdoelen van het project. Dat is je basis voor de lessen. **Wat je NIET in het werkboek vindt: Let op: je vindt GEEN papieren handleiding voor de montage!** Om direct aan de slag te gaan met **bouwen en programmeren**, hebben we **uitgebreide video's** voor je klaargezet. Dit is de snelste manier om je project tot leven te brengen.

Scan de QR-code en je krijgt direct toegang tot:

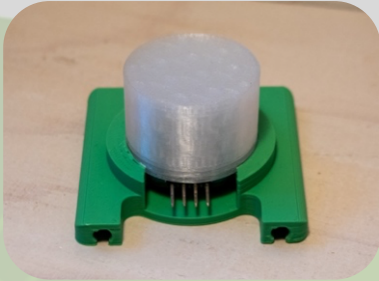
Bouwvideo's (Stap-voor-Stap): Zie precies hoe je elk component in elkaar zet, zonder gedoe. Perfect voor als je snel wilt weten hoe het in elkaar past.

Code Tutorials Leer hoe je de module programmeert met duidelijke instructies en voorbeelden, zodat je je eigen coole functies kunt toevoegen.

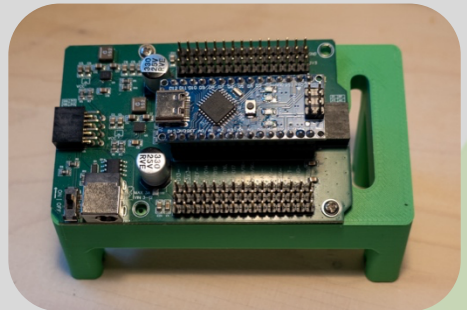
Veel succes met je project!

3-DOF manipulator

Andon Light



Conveyor belt

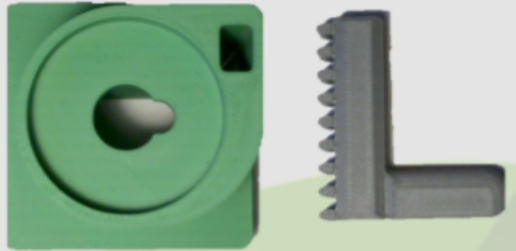
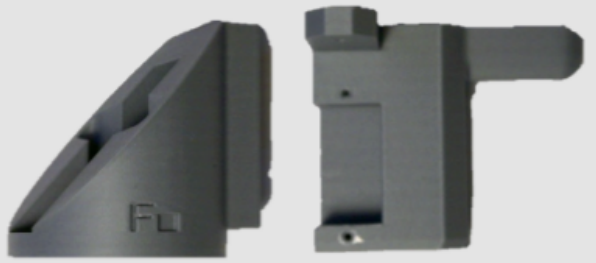


Mainmodule

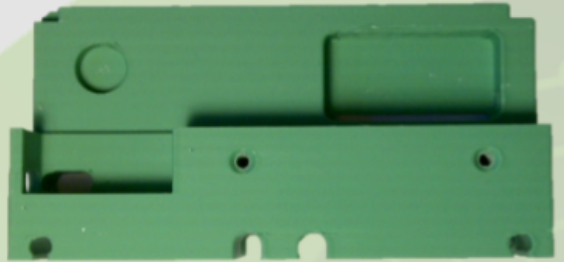
Check de Video's!



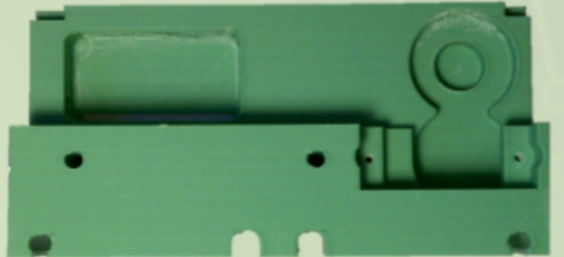
In the box



1x 4pins



4x 3pins



USB > USB-C



DC > USB-C

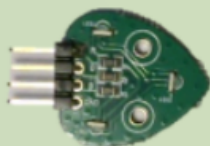




3x



1x



2x



4x



2x



1x



15x



4x

2x



2x

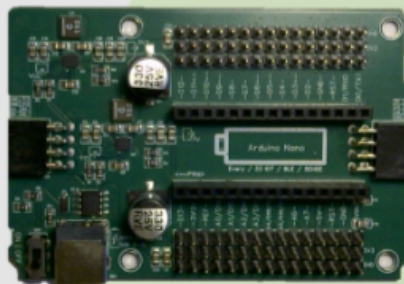
2x



2x



2x



28x



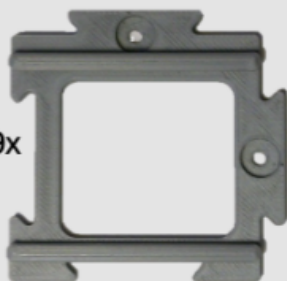
28x



1x

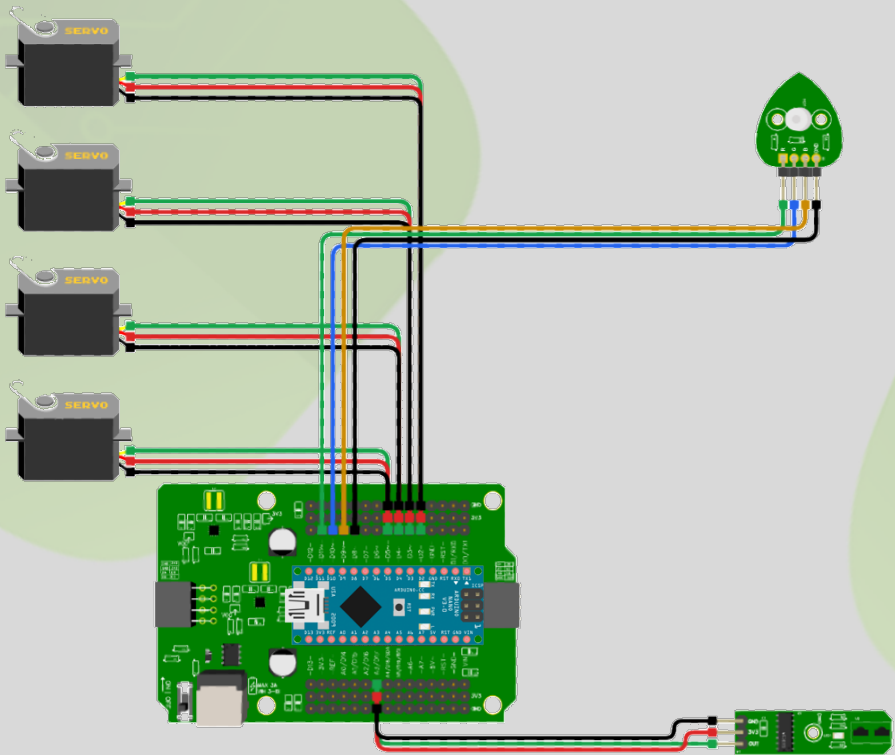


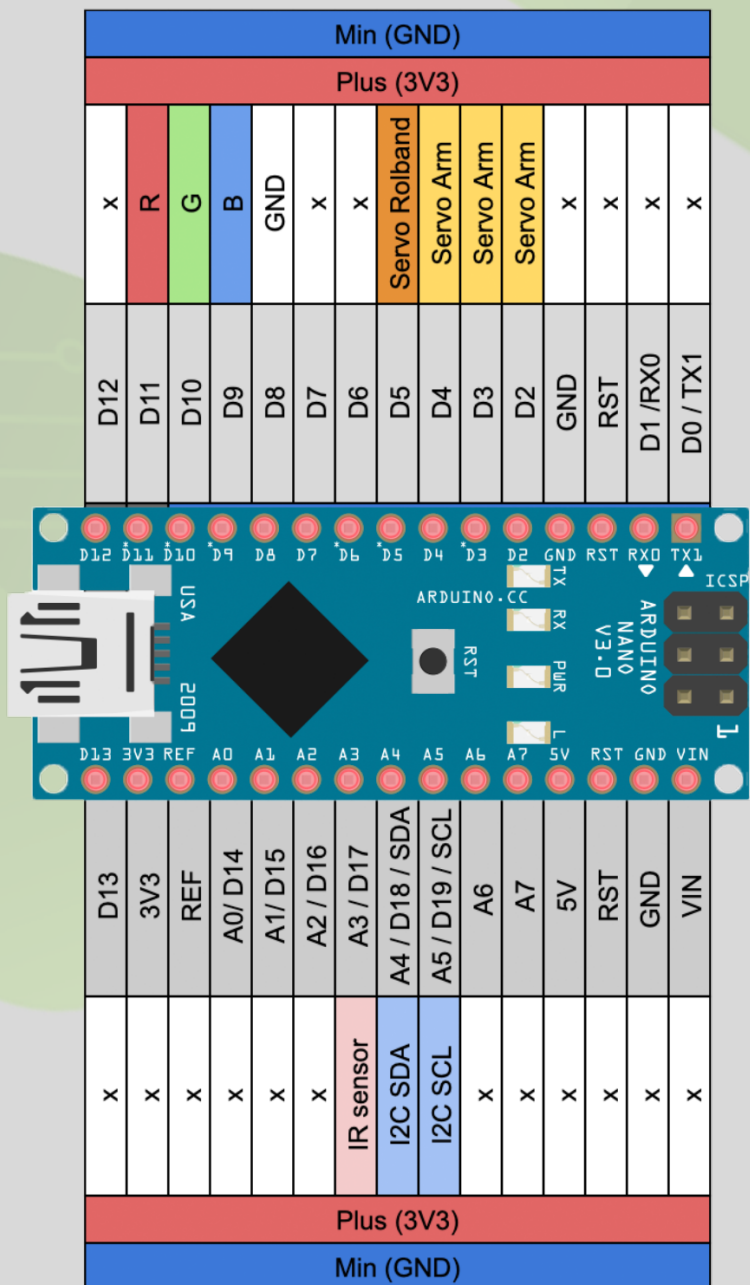
9x



Connection overview

In de **bouw filmpjes** wordt al uitgelegd hoe je de robot in elkaar zet en hoe je het aan moet sluiten. Dit allemaal samengevat vind je hier het **aansluitschema**.





A deep dive into Leaphy

Wil je echt de diepte in met de Arduino Nano en ontdekken hoe je code schrijft met de Leaphy Murphy shield? Check dan de officiële: ***Leaphy electronics & easybloqs Manual***.

Deze gids is dé bron om snel te leren werken met:

De Leaphy Shields:

Een helder overzicht van de hardware (de Delphy en Murphy shields). Deze shields zijn de uitbreidingen voor je Arduino Nano.

Sensoren en Actuatoren:

Alles over de drie soorten sensoren (digitaal, analoog en I2C) en actuatoren (DC-motoren en servo's). Je leert precies hoe je ze aansluit en programmeert.

Programmeren met Easybloqs:

Een stap-voor-stap uitleg over de Easybloqs software, die programmeren supertoegankelijk maakt via blokcode.



Leaphyeasybloqs.com



The Modular Base

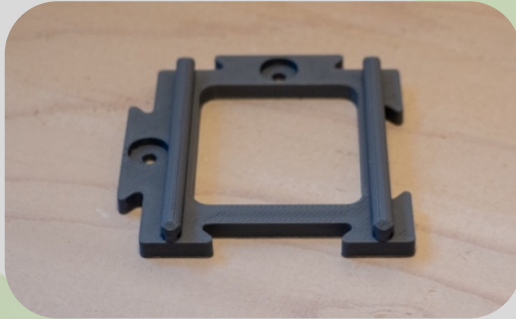
De **Modular Base** (de werkvloer) is een grote, flexibele puzzel voor de **modules** (machines). In de industrie noemen we dit **modulair werken**: elke module kun je overal plaatsen en vervangen.

De werkvloer (die bestaat uit de puzzelstukken) zorgt dat alle modules altijd op een vaste, gedefinieerde plek staan.

Zorg dat de modules stevig op de juiste **puzzelstukken** staan. Goede plaatsing is **belangrijk** voor de samenhang.

Test Je Kennis: *Waarom is het in een echte fabriek zo belangrijk dat machines modulair zijn? (Denk aan aanpassen, uitbreiden en repareren!)*

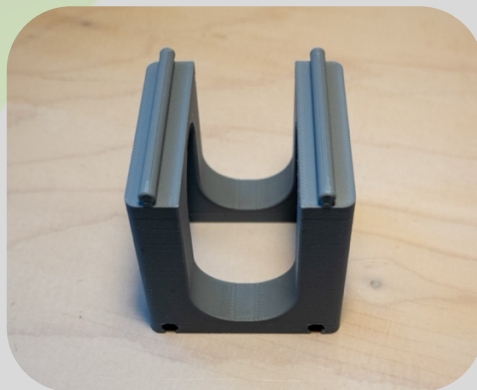
Base puzzelstuk



Basis werkvloer



Verhoging



The Control Hub

Om de fabriek te laten werken, gebruiken we de **Arduino Nano**. Dit is een **kleine computer** die net zo slim is als de grotere Arduino uno! Zie de Nano als het **mini-brein** dat je kunt programmeren voor al je projecten. Omdat de Nano zo klein is, past hij perfect in robots.

We klikken de Arduino vast op het **Leaphy Murphy Shield**. Dit extra bordje regelt de spanning en maakt het aansluiten van alle onderdelen veel makkelijker. Zo heb je veel minder losse draadjes nodig. Dit bordje wordt op de main module gemonteerd, zodat het op de werkvloer kan worden vastgezet.

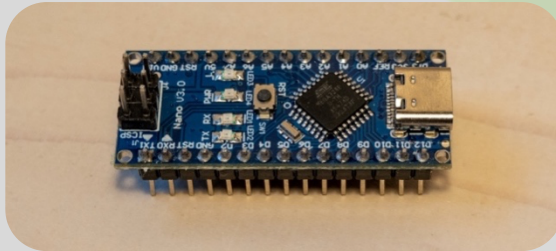
We programmeren de Arduino met **Leaphy EasyBloqs**. Dit is een eenvoudige, visuele programmeertaal die je online vindt op **leaphyeasybloqs.com**. Door blokken te slepen, schrijf je de code voor de fabriek of robot. Dit maakt het programmeren makkelijk.

De Arduino heeft **I/O-pinnen** (Input/Output). De pinnen zijn de metalen aansluitpunten op het bord. Ze zijn de communicatie-punten: Het ontvangt hier gegevens (Input, van sensoren) of verstuurt ze (Output, naar motoren of lampjes). Je schrijft de code op je computer en upload deze eenvoudig via een USB-kabel.

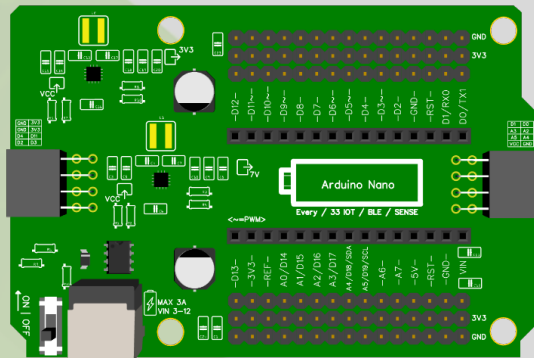
De pinnen kunnen **analoog** en **digitaal** zijn.

Digitaal is de simpelste communicatievorm: aan of uit. Denk aan een lichtschakelaar. Een digitaal signaal heeft slechts twee waarden: **HOOG (High, 1)**, wat 3 volt (AAN) betekent, of **LAAG (Low, 0)**, wat 0 volt (UIT) betekent. De digitale pinnen zijn geschikt voor simpele taken, zoals het aan- of uitzetten van een LED

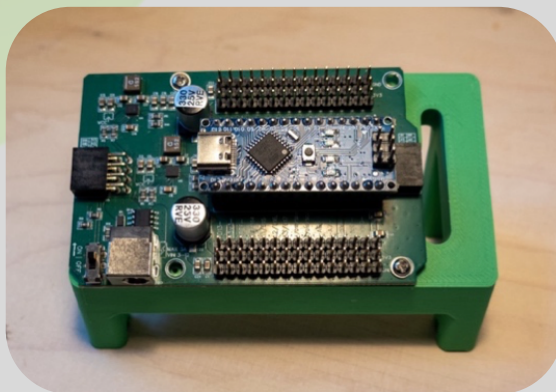
Arduino nano



Leaphy Murphy shield



Main module



Andon Light

In fabrieken noemen we een **statuslicht** een '**Andon light** of **Signaaltoren**. Dit is een soort verkeerslicht dat snel laat zien hoe het met een machine gaat. Rood betekent direct gevaar of een storing. Geel is een waarschuwing of aandacht nodig. Groen betekent dat alles veilig werkt en de machine klaar is.

De aansturing van het **Andon Licht** werkt via **Digitale Outputs** (de uitgangen van de Arduino). Elke kleur (rood, geel, groen) heeft een eigen pin.

Deze pin stel je in op **HIGH (1)** voor AAN (stroom) of **LOW (0)** voor UIT (geen stroom).

Om een kleur te kiezen, zet je de juiste pin op HIGH.

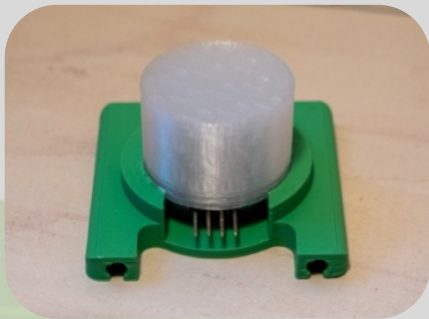
Kleur	Digitale Pin	Status
Rood	Pin 13	FOUT! Stop, of Wachten op een kritische actie.
Geel	Pin 12	OPLETTEN! Voorbereiding, of bezig met een lange cyclus.
Groen	Pin 11	START! Operationeel, alles is gereed.

Software

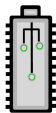
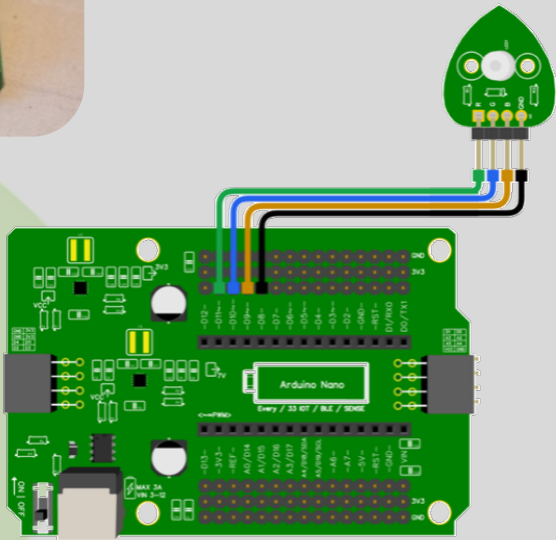
Ga naar: [Leaphyeasybloqs.com](https://leaphyeasybloqs.com)

Kies de Arduino nano tegel. Zoek onder actuatoren het volgende blokje en sleep deze in het Leaphy programmablok. In het voorbeeld wordt de groene LED op een heldere groene kleur: het rode deel wordt uitgeschakeld (waarde 0), het groene deel wordt op maximaal (waarde 255) gezet, en het blauwe blijft uit (waarde 0). Hierdoor brandt alleen groen.

Adon light



Aansluitschema Adon light



Arduino Nano



Actuatoren

Leaphy

LED Rood 0 Groen 255 Blauw 0

Waarde 0 LED uit
Waarde 255 LED Helderheid MAX

Sub-programs

In programmeeromgevingen zoals leaphyEasyblogs.com kun je complexe taken overzichtelijk maken door ze op te splitsen in kleine stukjes. Deze stukjes noemen we **subprogramma's** (of functies).

Met een eigen blokje maak je een lege functie aan. Je geeft de functie een duidelijke naam (hier staat "naam" als voorbeeld). In dit blokje plaats je de acties die uitgevoerd moeten worden wanneer je de functie later aanroept. Zonder opdrachten doet de functie nog niets, maar in de praktijk schrijf je hier een reeks robotacties die je met één enkele opdracht kunt laten uitvoeren.

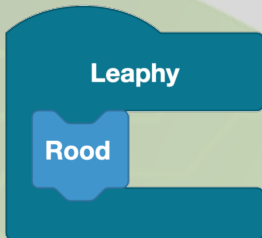
De aangemaakte sub programma's vind je terug in het menu onder het kopje '**Eigen Blokken**'.

De code gaat pas écht werken als het sub programma wordt aangeroepen in het Leaphy blok!

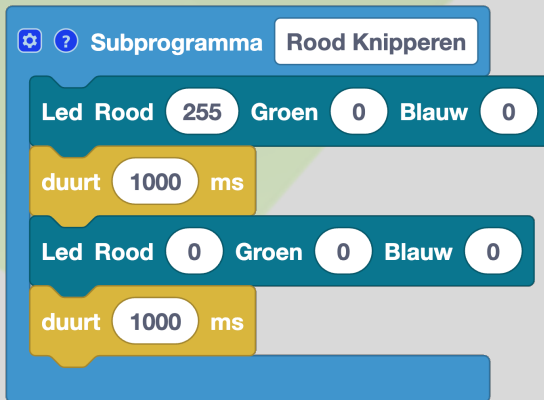
We willen graag in plaats van een vaste kleur een **signaal** of **waarschuwing** doorgeven. Dit doen we door een lampje te laten **knipperen**, net zoals je ziet bij waarschuwingslampen in een echte fabriek.

De cyclus voor de waarschuwing ziet er nu zo uit:

1. De rode LED gaat aan.
2. De LED moet één seconde (1000 milliseconden) zo blijven staan.
3. De rode LED gaat weer uit.
4. Hij moet daarna één volle seconde (1000 milliseconden) uitgeschakeld blijven staan.



Sleep de functie Rood in het leaphy blok.



* De LED is een **RGB LED** (rood, groen en blauw), wat betekent dat je kleuren kunt combineren om alle mogelijke kleuren te maken die je wilt.

The Conveyor Belt

De conveyor belt (transportband) is belangrijk voor je logistiek! Hij zorgt dat alle producten constant door je fabriek stromen. De band verplaatst items van A naar B.

Jouw Hoofdmodule (het 'brein') regelt de snelheid én de richting van de band.

Intelligentie: Slim Stoppen.

Hij is pas slim als hij stopt wanneer een product de exacte positie bereikt. Daarom is een reflectie sensor (de 'input') nodig. Die sensor zegt tegen de Hoofdmodule:

"Stop, het product is hier!"

Hardware

Continue Servo D5

Pin	Kleur kabel	
GND	Geel	GND
VCC	Zwart	3V3
PWM	Rood	<u>D5</u>

Infrarood Sensor D6

Pin	Kleur kabel	
GND	Geel	GND
VCC	Zwart	3V3
OUT	Rood	<u>D6</u>

Test Je Kennis: Wat voor type motor zou je gebruiken voor een transportband die af en toe moet stoppen en dan weer verder moet?

(Tip: Denk aan de **DC-motor** die alleen aan/uit kan, of de **Stappenmotor** die heel nauwkeurig kan bewegen.)

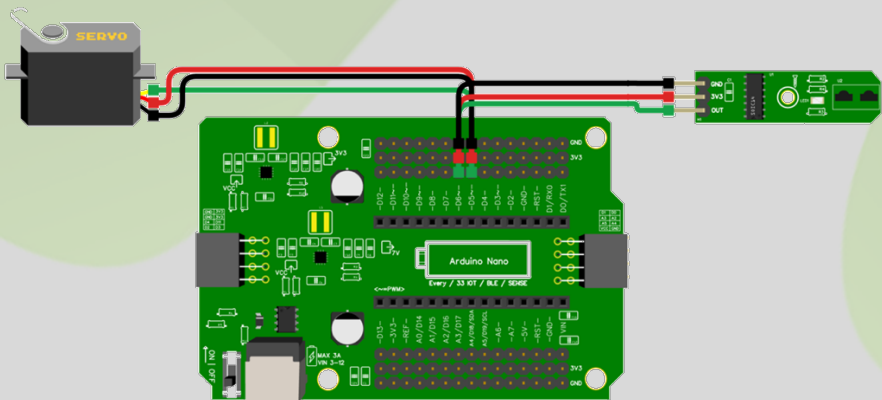
Conveyor Belt



Reflectie sensor



Aansluitschema Conveyor belt



Variables

Dit blok start de motor door één van de continue servomotoren aan te sturen.

De servo is ingesteld op D5 (digitale pin 5) op een neutrale waarde, die overeenkomt met **90°**. Met andere woorden: zodra het programma start, draait servo 5 **niet**

**notitie*

(Let op: Bij een continue servomotor stopt de motor meestal bij een waarde van $90^\circ \pm 1^\circ$. Waardes boven of onder 90° zorgen voor rotatie.)

Read Sensor

De robot kijkt op pin D6 (digitale pin 6) en leest of er een signaal aanwezig is op de reflectie sensor. Het gelezen getal (waarde 1 of 0) wordt vervolgens opgeslagen in een variabele die we “IR Sensor” noemen, zodat later in het programma kan worden gecontroleerd wat die ingang aangeeft.

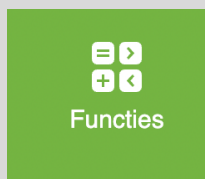
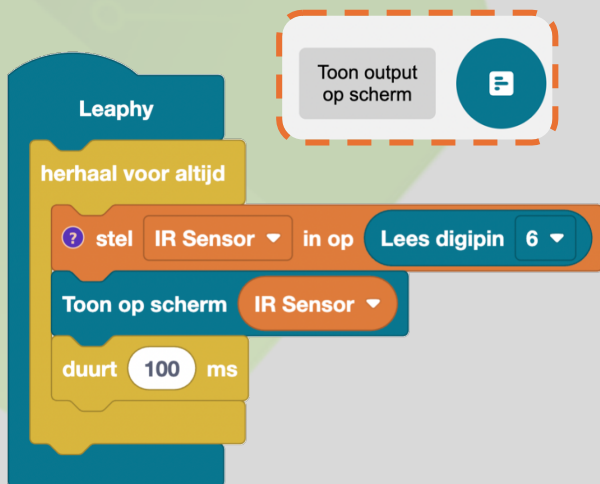
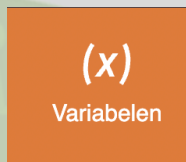
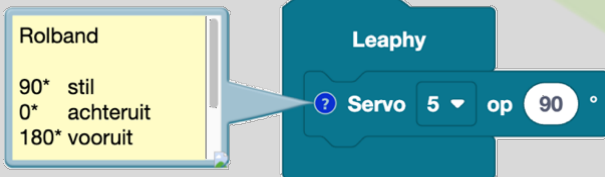
Output on screen

Toon op scherm, zorgt ervoor dat de robot het getal dat hij van de IR-sensor heeft gelezen, op het scherm laat zien. Met andere woorden: elke keer dat de code wordt uitgevoerd, wordt de huidige sensorwaarde (bijvoorbeeld 0, 1 of een ander getal) zichtbaar op het display, zodat je kunt zien wat de sensor meet.

Functions

Het groene blok “functies” kijkt naar de waarde die de sensor heeft opgegeven. Als die waarde precies 1 is, wordt er één set acties uitgevoerd; als de waarde iets anders is, wordt een andere set acties uitgevoerd.

Met andere woorden: er wordt een eenvoudige als-dan-beslissing genomen die alleen “waar” (waarde = 1) of “niet waar” (waarde $\neq$ 1) kan zijn.



IF Else

Dit blok is een **als-dan-anders**-structuur.

De hoofdmodule kijkt eerst naar een (nog niet ingevulde) voorwaarde.

- **Als** die voorwaarde waar is, voert hij de opdrachten uit die tussen de eerste accolades staan.
 - **Anders** (dus als de voorwaarde onwaar is) voert hij de opdrachten uit die tussen de tweede accolades staan.
-

Stel dat de module in een eindeloze lus verkeert: er worden steeds dezelfde handelingen herhaald. Bij elke herhaling wordt eerst de **infraroodsensor** uitgelezen die is aangesloten op de specifieke ingang D6 (digitaal pin 6). De sensor levert twee mogelijke waarden: 1 bij detectie van een obstakel of object en 0 als er geen detectie plaatsvindt.

- Als de sensor de **waarde 1** aangeeft, wordt de rolband (servo) die op pin D5 is aangesloten naar een hoek van 90° gestuurd.
- Anders, als de **waarde 0** aangeeft, roteert dezelfde motor naar rechts, wat neerkomt op: $90^\circ + 10^\circ = 100^\circ$.

Na het nemen van deze beslissing en het instellen van de servo, wordt de controle direct opnieuw gestart:

De sensor wordt opnieuw uitgelezen en de servo eventueel aangepast. Na het nemen van deze beslissing en het instellen van de servo, wordt de controle direct opnieuw gestart: de sensor wordt opnieuw uitgelezen en de servo eventueel aangepast.

Dit proces wordt oneindig herhaald, waardoor continu wordt gereageerd op de veranderingen die door de infraroodsensor worden waargenomen.



The 3-DOF Manipulator

De robotarm is een **3-DOF Manipulator** (Degrees Of Freedom). Dat betekent dat deze drie onafhankelijke bewegingen kan maken om een object in 3D-ruimte te bereiken: Roteren, Hoogte bepalen, en Grijpen.

Een robot doet precies wat jij zegt, stap voor stap. Dit noemen we **sequentiële besturing** of **robotkinematica**. Je moet elk onderdeel van de beweging uitschrijven: **Doel**: Pak het product op.

1. **Stap 1**: Roteer naar de transportband.
2. **Stap 2**: Laat de Pitch (Hoogte) zakken.
3. **Stap 3**: Sluit de Grijper.
4. **Stap 4**: Wacht kort.
5. **Stap 5**: Breng de Pitch (Hoogte) omhoog.
6. **Stap 6**: Roteer naar de afleverplek.

Denk als een Robot: Wat gebeurt er als je stap 2 en 3 omdraait? Juist, dan grijpt de robot in de lucht! De volgorde van de acties is alles.



As	Wat doet hij?	Type Output	Jouw Programmeerbereik
1. Rotatie	Draait de hele arm horizontaal.	Servo 180°	Draai hoek waarden (bv. tot)
2. Hoogte	Brengt de arm omhoog of omlaag.	Servo 180°	Hoekwaarden (bv. (omlaag) tot (omhoog))
3. Grijpen	Openen en sluiten van de gripper.	Servo 180°	Gripbreedte (bv. (dicht) tot (open))

Hardware

Servo Draaien D2

Pin	Kleur kabel	
GND	Geel	GND
VCC	Zwart	3V3
PWM	Rood	<u>D2</u>

Servo Hoogte D3

Pin	Kleur kabel	
GND	Geel	GND
VCC	Zwart	3V3
PWM	Rood	<u>D3</u>

Servo Grijper D4

Pin	Kleur kabel	
GND	Geel	GND
VCC	Zwart	3V3
PWM	Rood	<u>D4</u>

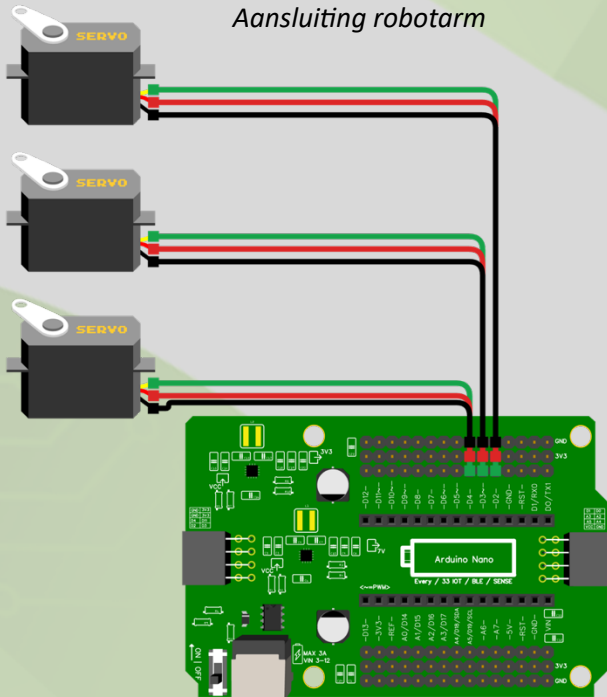
Software

De robotarm krijgt de opdracht om drie van zijn servomotoren één voor één in een bepaalde positie te zetten. Dit gebeurt razendsnel, waardoor het lijkt alsof alles tegelijk gebeurt:

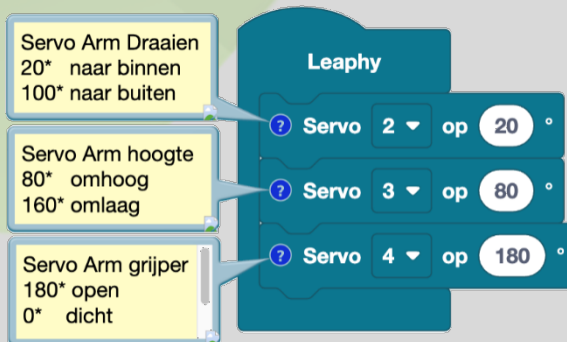
- Servo 2 draait naar een hoek van 125° .
- Servo 3 draait naar een hoek van 135° .
- Servo 4 draait naar een hoek van 180° . (180° is de maximale stand).

Wat betekent dit? Zodra de code wordt uitgevoerd, bewegen de genoemde motoren direct naar die specifieke hoeken.

Dit is vaak de eerste "setup" van de robot. Je gebruikt dit om onderdelen in een gewenste uitgangspositie te zetten. De robot staat dan in de perfecte starthouding voor de rest van zijn taak!



Software robotarm



*Uitdaging: Waarom zou je de servo's in de setup in een vaste hoek zetten?
(Hint: Denk aan de positie van de grijper en de arm voordat hij zijn eerste voorwerp pakt!)*

In de drie sub programma's staan drie 'functies' om de robotarm in een bepaalde positie te zetten.

- Servo 2 draait naar een hoek van 80°.
- Servo 3 draait naar een hoek van 160°.
- Servo 4 draait naar een hoek van 180°

Dit is vaak de eerste "setup" van de robot. Je gebruikt dit om onderdelen in een gewenste uitgangspositie te zetten. De arm staat dan in de perfecte starthouding voor de rest van zijn taak!

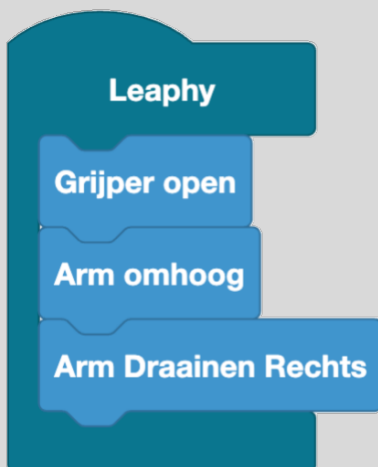
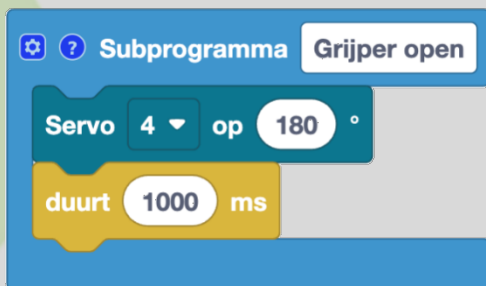
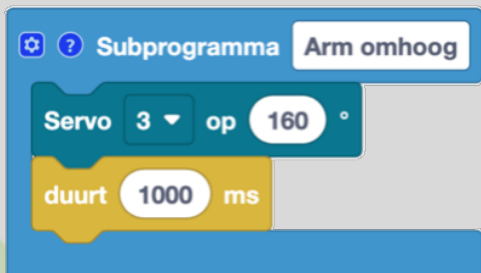
Zodra dit gedeelte van de code wordt uitgevoerd, bewegen de genoemde motoren **direct** naar die specifieke hoeken die je hebt ingesteld.

De code gaat pas écht werken als het sub programma wordt aangeroepen in het Leaphy blok!

Uitdaging: Met deze voorbeelden kun je nu ook sub programma's maken die precies het **tegenovergestelde** doen! Denk aan: Arm omlaag, Arm Draaien naar Links en de Grijper dicht. Zo bouw je je eigen robot-moves!



Sleep de sub programma's
in het leaphy blok.



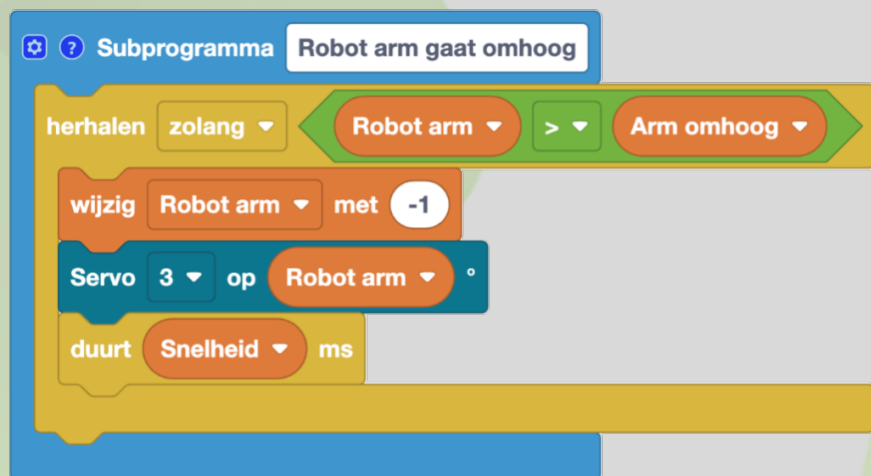
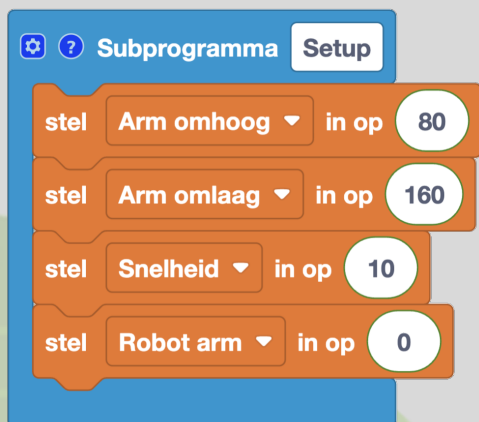
Je merkt dat de robotarm een erg snelle of schokkende beweging maakt naar zijn eindstand, die is ingesteld zoals 0 of 180 graden. Om de arm **gecontroleerd** en **soepeler** aan te sturen, is het beter om dit in kleine stapjes te doen.

In plaats van in één keer van 0 naar 180 graden te gaan, bewegen we van 0 naar 1, 2, 3, 4 enzovoort, met een kleine **vertraging** ertussen. Deze vertraging kunnen we zelf bepalen.

Het makkelijkst is om eerst een **sub programma** te schrijven, zoals hieronder. Dit sub programma vertelt de robot precies hoe hij de arm weer omhoog moet bewegen.

Sub programma: De Stappen

1. **Herhaling:** Blijf herhalen zolang de arm hoger staat dan de gewenste "omhoog positie". Met andere woorden: zolang de huidige hoek van de arm groter is dan de hoek die we als "omhoog" hebben ingesteld, blijft de robotarm doorgaan.
2. **Verkleinen:** Verklein de huidige hoek met één graad. Elke keer dat de lus wordt doorlopen, wordt de waarde die de arm beschrijft met 1 verminderd. Zo komt de arm langzaam naar de juiste positie.
3. **Aansturing:** Stuur het signaal naar servo motor D3. Nadat de hoek is aangepast, zet de robot servo motor D3 op de nieuwe hoek (in graden). Dit maakt dat de arm zich fysiek een graadje omhoog beweegt.
4. **Pauze:** Wacht even voordat de volgende stap wordt gedaan. De arm stopt kort, gedurende het aantal milliseconden dat onder "Snelheid" is ingesteld, zodat de beweging soepel verloopt en niet te snel gebeurt.



Let op: We gaan omhoog naar 0 graden en naar beneden naar 180 graden. Dit komt omdat de Servo op zijn kop is gemonteerd.

Sine Wave Control

Om een robotarm in een vloeiende beweging te laten bewegen—waarbij we rustig opstarten, versnellen en dan langzaam stoppen—kunnen we de servo aansturen met een sinusberekening.

Mocht dit een te grote stap zijn, kijk dan een paar pagina's verder en probeer het voorbeeldprogramma uit.

De **Y**-waarde is de hoek (in graden) van de motor en de **X**-waarde is de tijd of afstand.

Deze waarde kan variëren van **0 tot en met 180**. De Min en Max van de Servo aansturing. Het aanpassen van de sinusfunctie aan deze waarden is een beetje ingewikkeld, maar we leggen het hier stap voor stap uit.

Voordat we de motor instellen, kijken we naar de basis: de standaard sinusoïde $y = \sin(x)$

Deze functie beschrijft een vloeiende, repeterende **golfbeweging**. Deze basisgolf heeft de volgende kenmerken:

Amplitude (A): De golf gaat van -1 (minimum) naar 1 (maximum). De amplitude is dus **1**.

Evenwichtsstand (D): De middenlijn van de golf is de x-as ($y=0$).

Periode: Eén volledige golf duurt 2π (ongeveer 6,28) eenheden op de x-as.

Startpunt: De golf begint bij $x = 0$ op de evenwichtsstand en gaat dan omhoog. Maak een variabele X en vervolgens Y aan. Deze kunnen we later nog hernoemen:

(x)

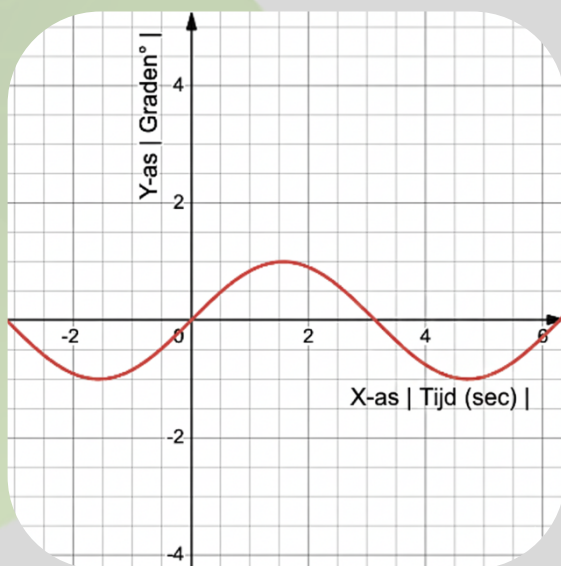
Variabelen

Geef je variabele een naam



Functies

Zoek de SIN functie



$$y = \sin(x)$$



Range and equilibrium (A & D)

De functie $y = \sin(x)$ is niet genoeg om dit voor elkaar te krijgen.

We gebruiken de algemene formule van de sinus:

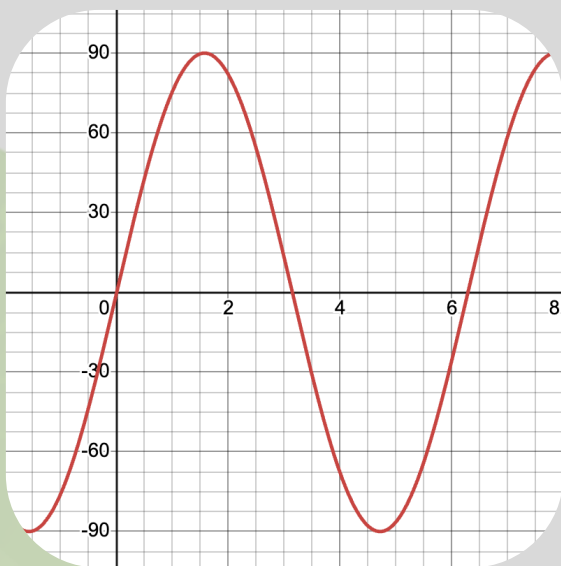
$$y = A \sin(B(x - C)) + D$$

Deze transformeren naar de beweging die we nodig hebben. We willen een totale draai van **180 graden** (van 0° naar 180°).

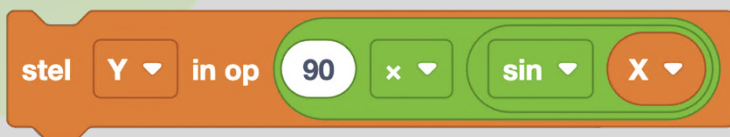
Stap 1. Amplitude (A): De amplitude is de helft van het bereik.

$$A = \frac{\text{Maximum} - \text{Minimum}}{2}$$

$$A = \frac{180 - 0}{2} = 90$$



$$Y = 90 \sin(x)$$



Stap 2: Evenwichtsstand (D): Dit is het midden tussen het minimum en maximum.

$$D = \frac{\text{Maximum} + \text{Minimum}}{2}$$

$$D = \frac{180 - 0}{2} = 90$$

De basisformule wordt dan:

$$y = 90 \sin(x) + 90$$

Resultaat: Deze grafiek heeft een bereik van

$$90 \sin(x) + 90:$$

Minimum = 0 en *Maximum* = 180

Zoek de volgende functies op en combineer de blokken door ze in elkaar te slepen.

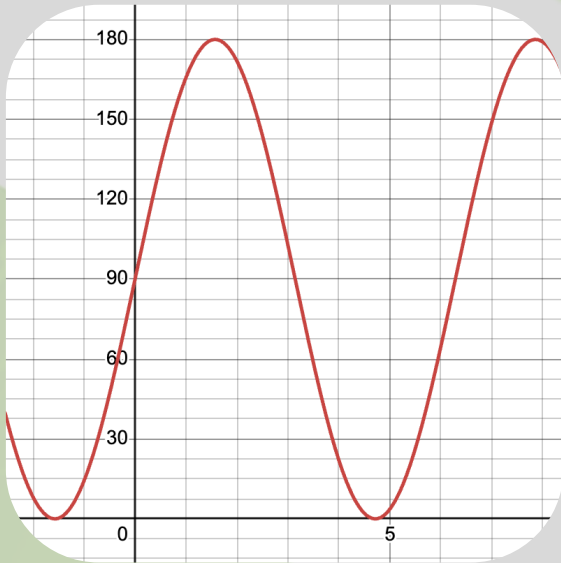
1

×

1

sin

123



$$y = 90 \sin(x) + 90$$

stel Y in op 90 × sin X + 90

Period and the Frequency (B)

De **periode** is de lengte van één volledige golf. Omdat we de motor willen laten draaien van **0 tot 180 graden** in één halve cyclus van de golf, moeten we de periode zo instellen dat de halve periode gelijk is aan de totale **tijd** (de lengte op de x-as) die de motor nodig heeft.

Laten we aannemen dat de draaibeweging **1 seconde** of **1 x-eenheid** duurt. Dit betekent dat de **halve periode** gelijk moet zijn aan 1. De volledige periode is dan **2**.

De formule voor de frequentiefactor B is:

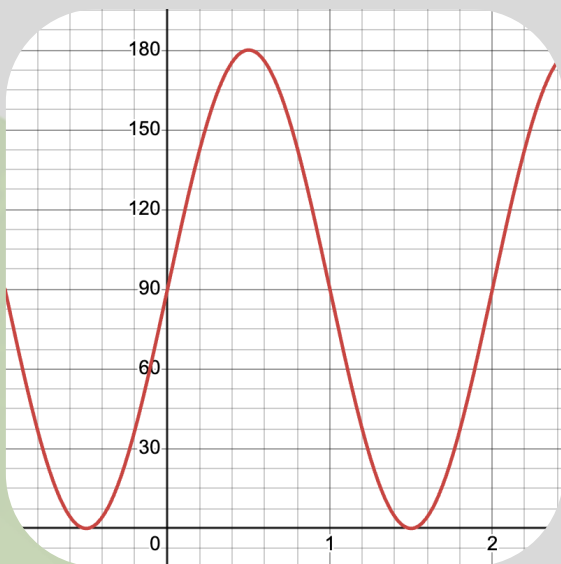
$$B = \frac{2\pi}{\text{Periode}}$$

Aangezien de gewenste Periode = 2:

$$B = \frac{2\pi}{2} = \pi$$

De formule is nu:

$$y = 90 \sin(\pi * x) + 90$$



$$y = 90 \sin(\pi * x) + 90$$

stel Y ▼ in op 90 x ▼ sin ▼ 3.14 x ▼ X ▼ + ▼ 90

The Phase Shift (C)

De grafiek snijdt de y-as nu nog op de evenwichtsstand ($y = 90$). Dit betekent dat de motor direct naar 90° zou springen. We willen echter dat de motor bij $x = 0$ (start) op **0 graden** (het minimum) begint, zodat de motor vloeiend kan starten met versnellen.

Hiervoor verschuiven we de golf horizontaal (**faseverschuiving C**). Omdat we willen dat de golf begint in het minimum, moet de grafiek een **kwart periode** naar rechts verschuiven. Een kwart van de periode (2) is **0,5**.

Je verschuift de golf 0,5 naar rechts door de x te vervangen door $(x - 0.5)$.

De uiteindelijke formule die het gewenste gedrag beschrijft, is:

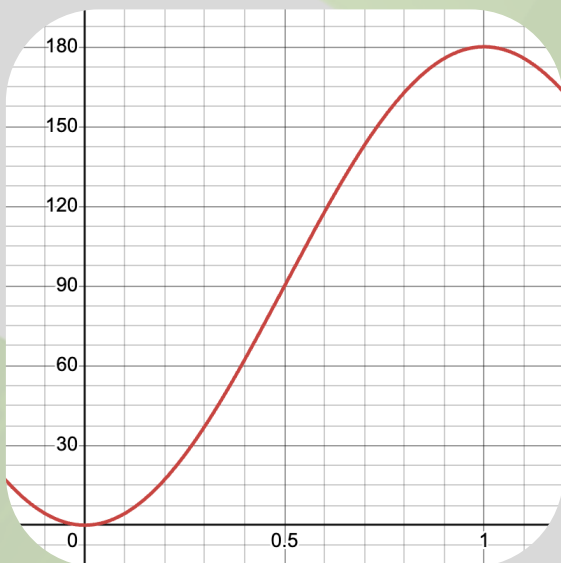
$$y = \sin(\pi(x - 0.5)) + 90$$

of, na distributie van $B = \pi$:

$$y = \sin(\pi x - 0.5\pi) + 90$$

Controle:

De grafiek toont precies dit gedrag: de golf begint op $x = 0$ en $y = 0$ en bereikt zijn maximum van $y = 180$ bij $x = 1$. Dit is de juiste



$$y = 90 \sin((x - 0.5) * \pi) + 90$$



Zelf een grafiek maken, check:



<https://www.desmos.com/calculator/>

Stel de variabele '**hoek**' in op $Gradenhoek = 90 * \sin((tijdstep - 0.5) * 3.14) + 90$

We voegen een '**herhaal zolang**'-blokje toe. Dit zorgt ervoor dat de motor beweegt zolang de **tijdstap** (de X-waarde) de eindwaarde (in dit geval **1**) nog niet bereikt heeft.

1. Als de eindwaarde nog niet bereikt is, **bereken** je de nieuwe waarde voor '**hoek**' met de vergelijking.
2. Daarna stuur je deze berekende waarde naar de **motor**.
3. Vervolgens tel je een kleine stap, bijvoorbeeld **0,05**, op bij de variabele '**tijdstap**'.
4. Wacht je **10 milliseconden** en dan begint het proces weer opnieuw!

Zodra de gewenste waarde bereikt is, moet je de variabele '**tijdstap**' terugzetten naar **nul**. Dan is die variabele klaar om de volgende keer opnieuw te starten.



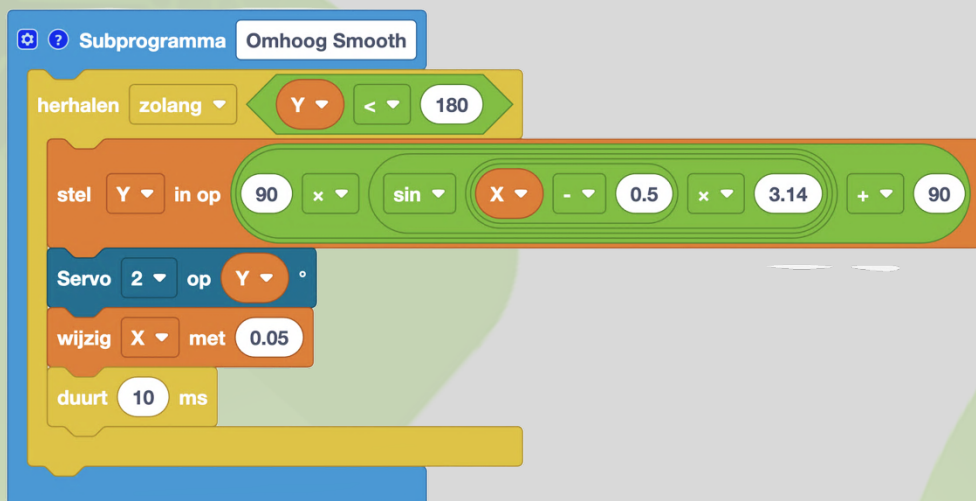
Snel Aanpassen

Je past de sinusformule ($y = A \sin(B(x - C)) + D$) aan door de variabelen **A**, **D**, en **B** te wijzigen.

1. **Andere Draaihoek** (bijvoorbeeld van 45° naar 135°):
 - o **A** wordt de helft van de draaiing: $(135 - 45) / 2 = 45$.
 - o **D** wordt het middelpunt: $(135 + 45) / 2 = 90$.
2. **Andere Kant Op Draaien** (bijvoorbeeld 180° naar 0°):

Maak de **Amplitude (A)** negatief. Gebruik $A = -90$.

3. **Tijd Aanpassen** (langzamer): Maak de **Frequentiefactor (B)** kleiner. Als de beweging 4 keer langer moet duren, deel je de B-waarde (3.14) door 4.



Now it's up to you



De belangrijkste onderdelen van het Leaphy Fabriek-project zijn doorlopen. Vanaf de hardware tot de code:

1. **De Componenten** De aansturing van de robotarm, de transportband, en het Andon Licht is bekend. Er is geleerd dat de Arduino Nano het programmeerbare brein is dat op het Leaphy Murphy Shield klikt om alle componenten (via I/O-pinnen) aan te sturen.
2. **De Logica** De aanpak om complexe taken op te delen in subprogramma's (functies) is bekend. Dit is belangrijk voor het overzichtelijk houden van code en om de robot een reeks acties meerdere keren te laten uitvoeren.
3. **De Beweging** Er is gezien hoe een schokkerige beweging kan worden vervangen door een vloeiende, gecontroleerde beweging. Er is geleerd om de wiskunde achter de sinusoïde te gebruiken om de motor te laten versnellen en afremmen.
4. **De Uitdaging** Nu bekend is hoe inputs (sensoren) kunnen worden uitgelezen en outputs (motoren, lampen) kunnen worden aangestuurd, is het tijd om de elementen zelf te combineren. De grootste uitdaging in de automatisering is het schrijven van de logica die de fabriek autonoom laat werken.

Er moet worden nagedacht over de stappen die nodig zijn om een product automatisch op te pakken en te verplaatsen:

- Hoe wordt gewaarborgd dat de transportband stopt als de sensor een product detecteert?
- Welke Als...DAN...ANDERS structuur wordt gebruikt om de robot de juiste actie te laten uitvoeren?
- Wat is de precieze volgorde van de bewegingen die de robotarm moet maken (Roteren, Hoogte, Grijpen)?

De bouwvideo's zijn beschikbaar om te zien hoe alles fysiek in elkaar zit, en de theorie is behandeld in dit werkboek. Er kan nu zelfstandig geprogrammeerd worden in Leaphy EasyBloqs.

Succes!

Software Explanation

Setup laden

Eerst worden een paar belangrijke waardes ingesteld, zoals hoe hoog de arm moet gaan, hoe ver hij naar links of rechts draait en hoe breed de grijper moet openen en sluiten.

1. **Beginpositie van de robotarm**

De arm, de draaibeweging en de grijper worden meteen naar hun startposities gezet, zodat alles vanaf een vast beginpunt start.

2. **Voor altijd blijven herhalen**

Daarna begint een eindeloze lus – de robot blijft dit steeds opnieuw doen totdat hij uitgeschakeld wordt.

- a) **Lopende band aan** – De band die dingen voortduwt wordt aangezet.
- b) **Groen lampje** – Een LED wordt groen, zodat je ziet dat het systeem klaar is.
- c) **Sensor controleren** – De robot kijkt of er een signaal (waarde 1) van een sensor op pin 17 binnenkomt. Deze sensor kan bijvoorbeeld een object op de lopende band detecteren.

3. **Als er iets gedetecteerd wordt (sensor = 1)**

Dan gebeurt er een hele reeks acties:

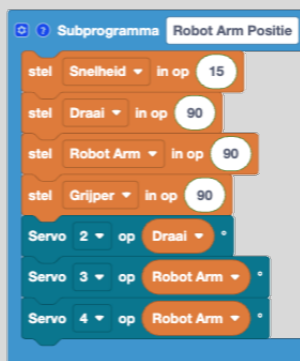
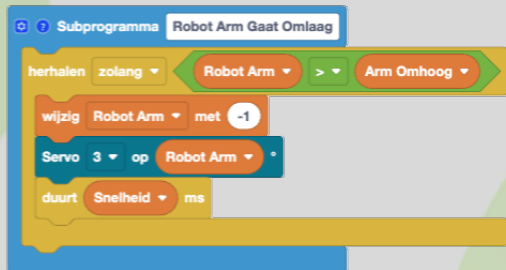
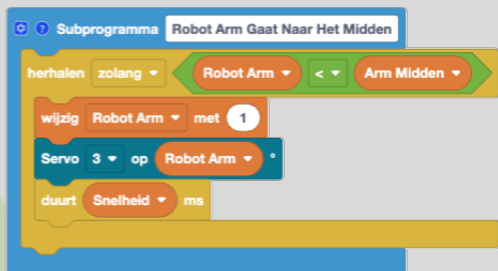
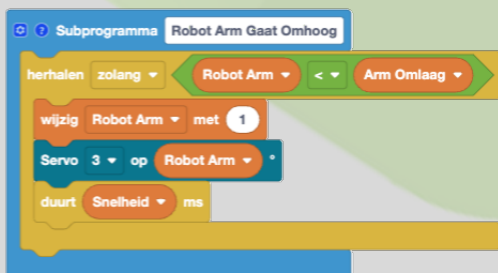
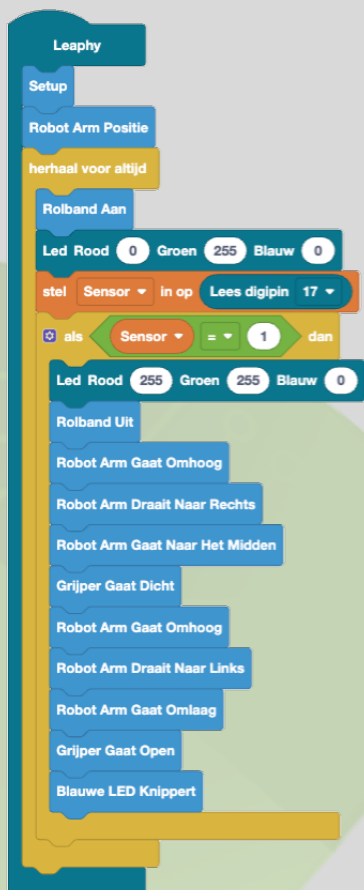
- d) **LED-geel** – Het lampje wordt geel om aan te geven dat er een object is gezien.
- e) **Lopende band uit** – De band stopt, zodat het object niet meer verder beweegt.
- f) **Arm omhoog** – De robotarm tilt zich omhoog.

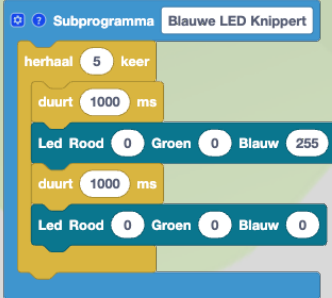
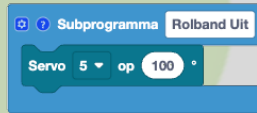
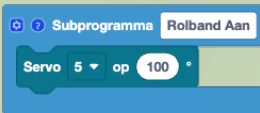
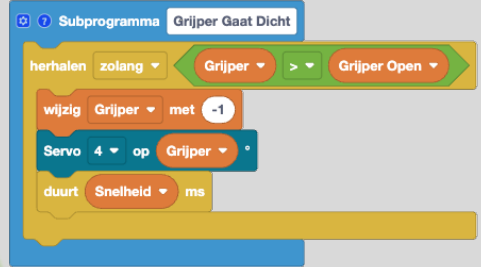
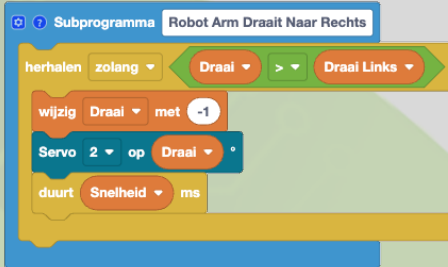
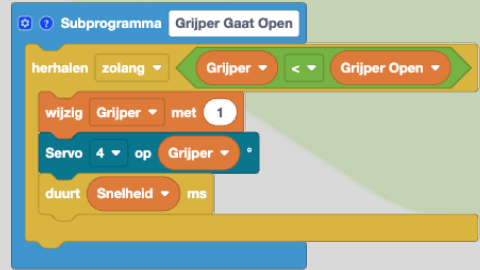
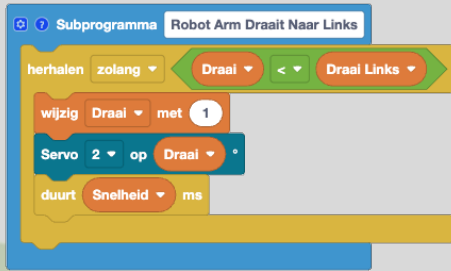
- g) **Arm draait naar rechts** – Vervolgens draait hij naar rechts.
- h) **Arm naar het midden** – Daarna beweegt hij zich naar de centrale positie.
- i) **Grijper sluit** – De grijper knijpt dicht om het object vast te pakken.
- j) **Arm omhoog (nog een keer)** – De arm tilt nog een keer omhoog, nu met het object erin.
- k) **Arm draait naar links** – Hij draait nu naar links om het object naar de gewenste plek te brengen.
- l) **Arm omlaag** – De arm laat het object langzaam zakken.
- m) **Grijper opent** – De grijper opent weer, waardoor het object losgelaten wordt.
- n) **Blauw lampje knippert** – Tot slot knippert een blauw lampje een paar keer om te laten zien dat de volledige handeling succesvol is afgerond.

4. **Daarna begint de cyclus opnieuw**

De robot gaat weer terug naar het begin van de lus, zet de band weer aan en wacht op het volgende object.

Kort samengevat: het programma houdt de robotarm en de lopende band continu in de gaten, wacht tot een object wordt gedetecteerd, pakt het object op, verplaatst het naar een andere positie, laat het los en geeft met een knipperende blauwe LED een signaal dat de taak voltooid is. Dit alles gebeurt steeds opnieuw, automatisch en zonder tussenkomst van een mens.





Complete Software

Upgrades

Om de fabriek nog slimmer en realistischer te maken, zijn er diverse **uitbreidingen** beschikbaar. Met deze extra modules, zoals de **OLED** en de **Pusher**, verhoog je de complexiteit en precisie van de automatisering.

De **extra Infrarood (IR) sensor** is beschikbaar als handige **uitbreiding**. Met deze sensor kan de robot nauwkeuriger producten detecteren en sorteren. Zo wordt de betrouwbaarheid van het hele geautomatiseerde systeem verhoogd.

Het **OLED-schermpje** is een handige add-on om data te tonen. Het is een compact display dat helder oplicht. Je kunt er gemakkelijk vijf regels tekst op laten zien, perfect voor het weergeven van de status of sensorwaarden.

De **Pusher** is een uitbreiding die schijven van de transportband schuift. De robot kan hiermee producten tijdelijk opzij zetten als een buffer, of ze sorteren. Dit verbetert de logistiek in de fabriek.

De **Turntable** (draaischijf) is een module die de schijven op de band kan draaien. Je gebruikt deze om de oriëntatie van een product te veranderen, bijvoorbeeld om het 90 graden te keren voor de volgende stap.

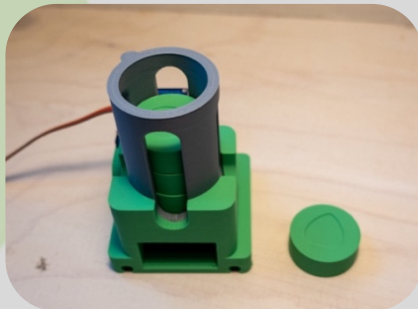
IR Sensor



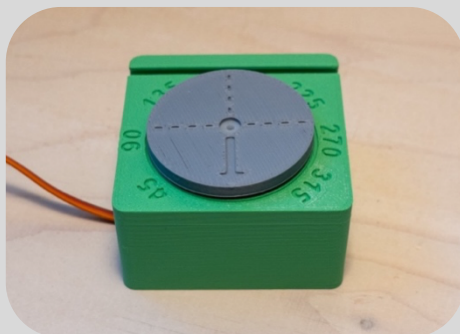
OLED-screen



Pusher



Turntable



Future Leaphy Factory

Bij Stichting Leaphy geloven we dat techniek nooit af is. Net zoals een echte fabriek constant wordt gemoderniseerd, staat ook de ontwikkeling van de **Leaphy Factory** nooit stil. We werken continu aan nieuw lesmateriaal om de wereld van robotica groter en uitdagender te maken.

Nieuwe Designs en Levels Achter de schermen ontwikkelen we nieuwe fysieke designs en "extra levels" voor de modulaire werkvloer. Waar je nu leert werken met de 3-DOF Manipulator en de conveyor belt, komen er in de toekomst complexere modules bij die modulair aansluiten op de fabriek.

Innovatieve Functies Ook de software van Leaphy EasyBloqs blijft groeien. We voegen regelmatig nieuwe programmeerfuncties toe, zodat je nog meer controle krijgt over sensoren en actuatoren. Of het nu gaat om slimmere variabelen of geavanceerde sub programma's, we zorgen dat de laatste versie van dit boekje altijd aansluit bij de nieuwste technologie.

Jouw Reis Gaat Verder

Dit werkboek is pas het begin. Houd onze website en de QR-codes in de gaten voor de **laatste updates, nieuwe video-tutorials en extra opdrachten**. Of je nu droomt van een carrière in de techniek of gewoon graag coole machines bouwt: bij Leaphy zorgen we dat je altijd een volgend niveau hebt om naar toe te groeien.

Blijf bouwen, blijf testen en vooral: blijf ontdekken!

Wist je dat? We luisteren naar de feedback van leerlingen en docenten om onze handleidingen en video's nóg duidelijker te maken. Jouw ervaring helpt ons om het lesmateriaal van morgen te bouwen.

LEAPHY 