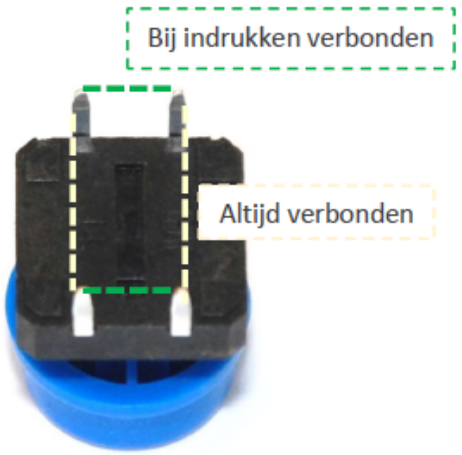
Leaphy Original - Level 8 - Drukknop

Veel apparaten werken met drukknoppen: één druk op de knop en het apparaat gaat aan en met nog een druk op knop gaat hij weer uit. Of: met één knopje kun je langs verschillende functies klikken.
In dit level sluit je zelf een drukknop aan op je Leaphy en leer je ermee werken.

Level 8.1 – Aansluitpinnen drukknop
Een schakelaar houdt de stroom tegen, totdat erop gedrukt wordt. Dan begint de stroom te lopen en daar kun je Leaphy op laten reageren.

Het bij Leaphy gebruikte drukknopje heeft vier pinnen. De pinnen tegenover elkaar zitten altijd aan elkaar vast (de gele stippellijntjes). Bij indrukken worden de pinnetjes die naast elkaar gemonteerd zitten ook mét elkaar verbonden (groene stippellijntjes). Daarmee zijn dus alle pinnetjes met elkaar verbonden.



Level 8.2 – Weerstand gevraagd
Arduinopoorten zijn heel gevoelig: het kleinste beetje stroom wordt al gezien als een signaal. Je zou het 'spookstroom' kunnen noemen. De Arduino denkt dan dat de digitale poort waarde 1 heeft, terwijl de knop al is losgelaten. Dat is lastig programmeren.

Dit los je op door de drukknop met behulp van een weerstand aan te sluiten.

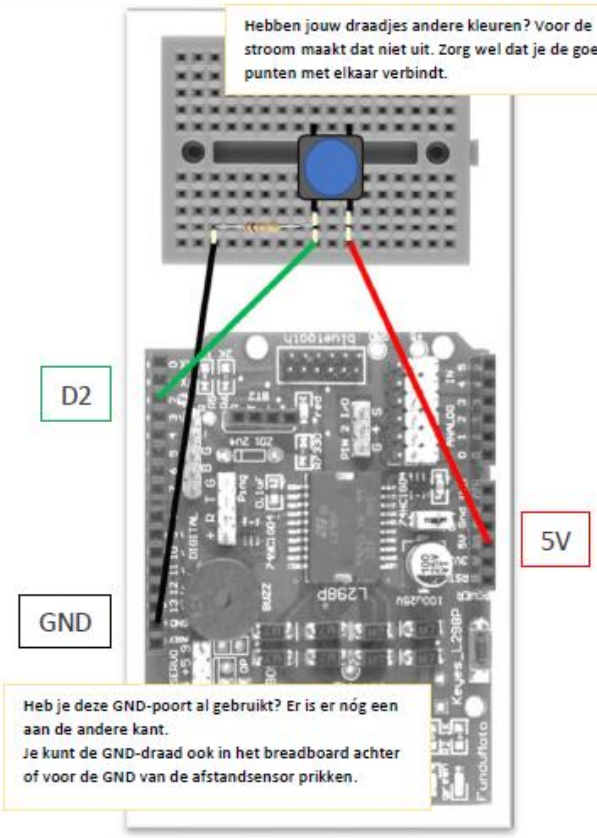
Dit is trouwens niet nieuw op je Leaphy: op de lijnvolgers die je eerder hebt gebruikt, zit ook al een mini-weerstandje ingebouwd. Bij de drukknop ga je die dus zelf toevoegen. Hoe dat gaat, lees je in level 8.3

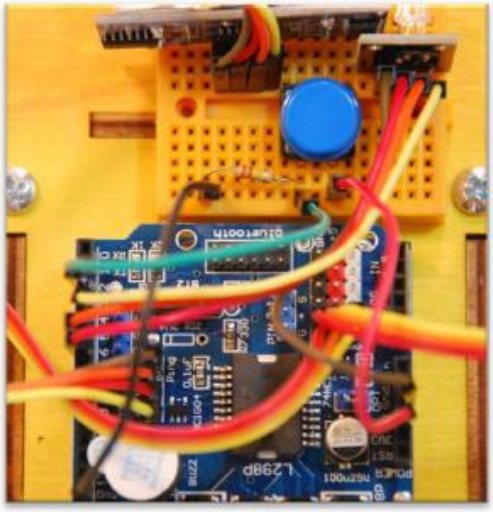
Level 8.3 – Drukknop-met-weerstand

In dit voorbeeld gebruiken we als de Arduinopoort *Digitale pin 2*.
Sluit die aan op één pin van de drukknop (groen).
Sluit de andere drukknop-pin aan op de 5V-poort (rood).
Als de knop nu wordt ingedrukt, zal de stroom vanuit V5 naar de D2-poort gaan en die zal een 1 uitlezen. Als je hem weer loslaat, stopt de stroom. Maar: de kans groot dat de poort op 1 blijft staan: dat is de spookstroom uit level 7.2. Die blijft hangen in poort D2.
Die moet je afvoeren naar de Ground.
Dat doe je door via het breadboard een verbinding te maken tussen de D2-pin en de GND. Kan dat zomaar?
Nee! Want bij het indrukken van de knop zou je dan de 5V ook direct met de GND verbinden: kortsluiting en oververhitting☹.
Dus is de *weerstand* nodig. Deze zet je tussen de D2-pin-kabel en de GND-kabel.

Is de knop niet ingedrukt? Dan zal eventuele spookstroom uit D2 via de weerstand naar de GND vloeien. De weerstand laat maar weinig stroom door, maar spookstroom is ook maar heel weinig. Pin D2 staat dan netjes op 0.
De weerstand noem je daarom een pull-down-resistor: hij trekt de waarde naar 0.

Als je de knop wél indrukt, kan de stroom vanuit V5 opeens twee routes kiezen:
Eén makkelijke weg, direct naar signaalpin D2.
Eén moeilijke weerstand-weg naar de GND.
Stroom kiest de makkelijkste weg en gaat dus naar pin D2: die leest een 1. De stroom gaat niet naar de GND en er is dus geen kortsluiting.

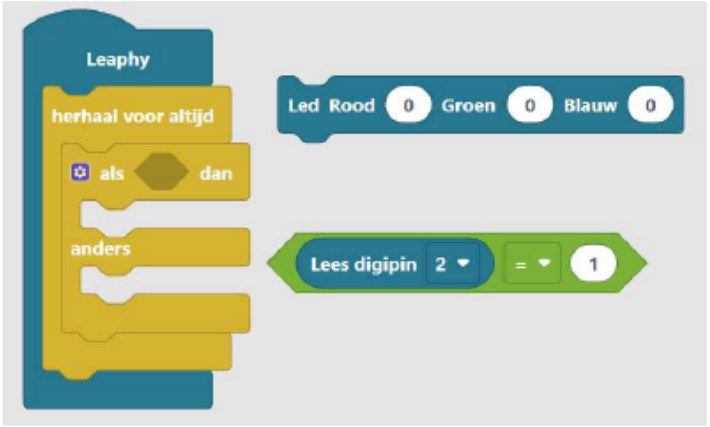




Level 8.4 – Eerste test

Test je drukknop met een simpel programma:
Ingedrukte knop = lichtje aan.
Losgelaten knop = lichtje uit.

Werkt het? Door naar level 7.5!



Level 8.5 – Aan en uit

Met één en dezelfde knop een ledje aan en uit zetten vraagt om een handige programmeertruc. Die leer je in dit level.

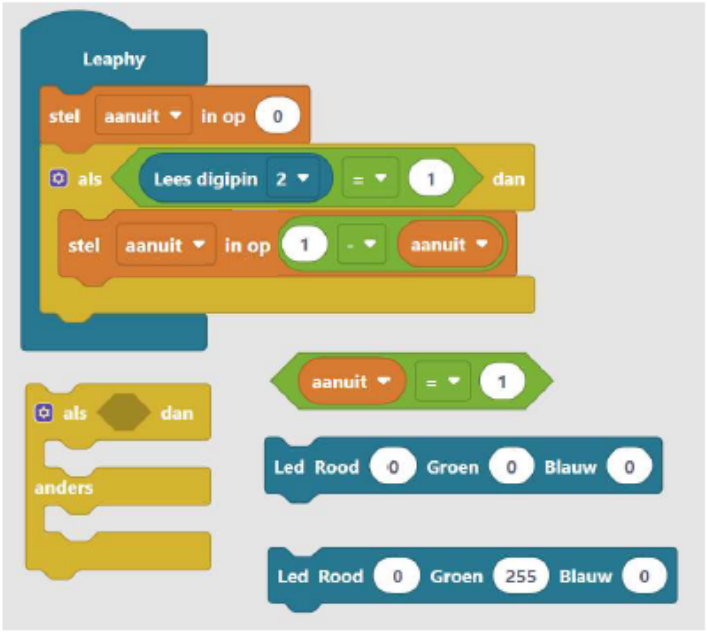
Maak een programma waarbij:
Eén keer drukken = led aan.
Nog een keer drukken = led uit.

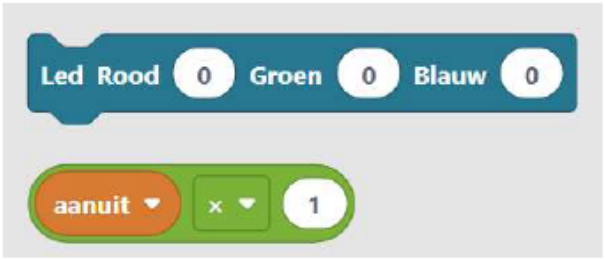
Maak hiervoor een variabele aan zoals hiernaast. Kijk even of je de rekentruc snapt! Door de variabele af te trekken van het getal 1, zal hij bij iedere druk op de knop wisselen van 1 naar 0 en van 0 naar 1.
Die 0 en die 1 kun je vervolgens gebruiken om je ledje aan en uit te zetten.

Dat kan met Als-Dan-Anders, zoals hiernaast.

Het kan ook met een vermenigvuldigingsopdracht. Dat scheelt programmeerblokjes en is iets interessanter qua wiskunde.

Werkt het....een beetje? In level 7.6 ontdek je hoe je knop het nog beter gaat doen!

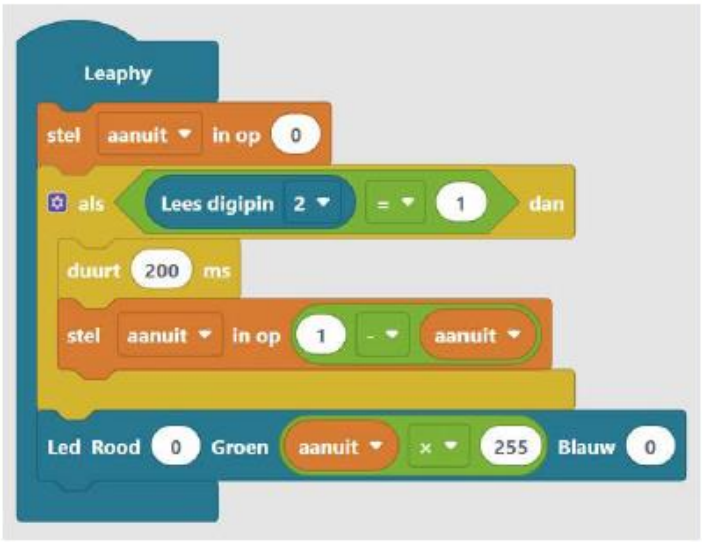




Level 8.6 – Denderstress

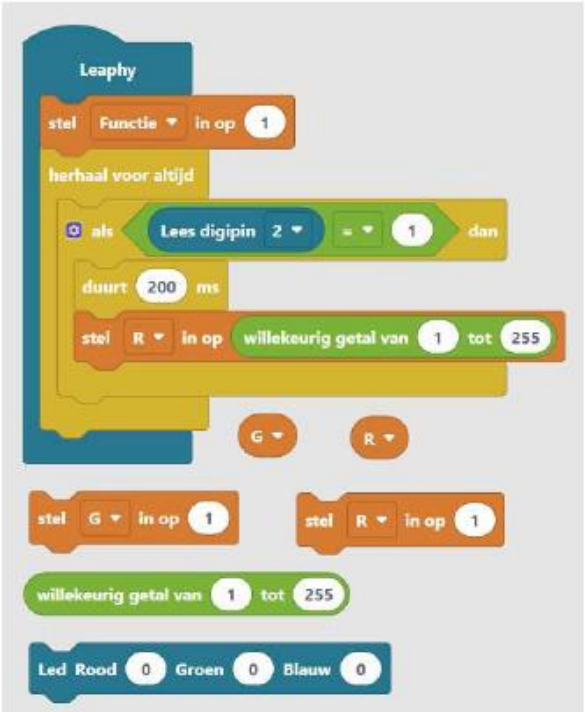
De knop blijkt niet heel betrouwbaar. Dat komt doordat de Arduino zo snel is. Bij het allereerste contact springt de poort op 1, maar 0,0001 seconde later is er nog steeds contact en dan misschien weer even niet en dan weer wel en dan nog steeds. De variabele springt ondertussen heel snel heen en weer tussen 1 en 0. En dat dus heel vaak en heel snel. Dit noem je het 'denderen' van een schakelaar.

Je kunt deze denderstress verhelpen door na het eerste contact een korte wachttijd in te lassen. Probeer maar wat voor jou werkt.



Level 8.7 - Willekleurig

Maak een programma dat bij iedere druk op de knop een willekeurige kleur maakt. Maak hiervoor drie kleurvariabelen:
R, G en B.
Waarom tussenvariabelen gebruiken?
Als je in het LED-blok drie groene willekeurblokken plakt, krijg je wel een heel breed ledjesblok...
(Dat is het nadeel van blokjescode...)



Level 8.8 – Meerdere functies I

Met één knopje langs verschillende functies scrollen: handig. Hier leer je hoe dat werkt.

Maak een nieuwe variabele: 'Functie'.

Maak daarmee een programma dat bij iedere druk op de knop de led een volgende kleur geeft.

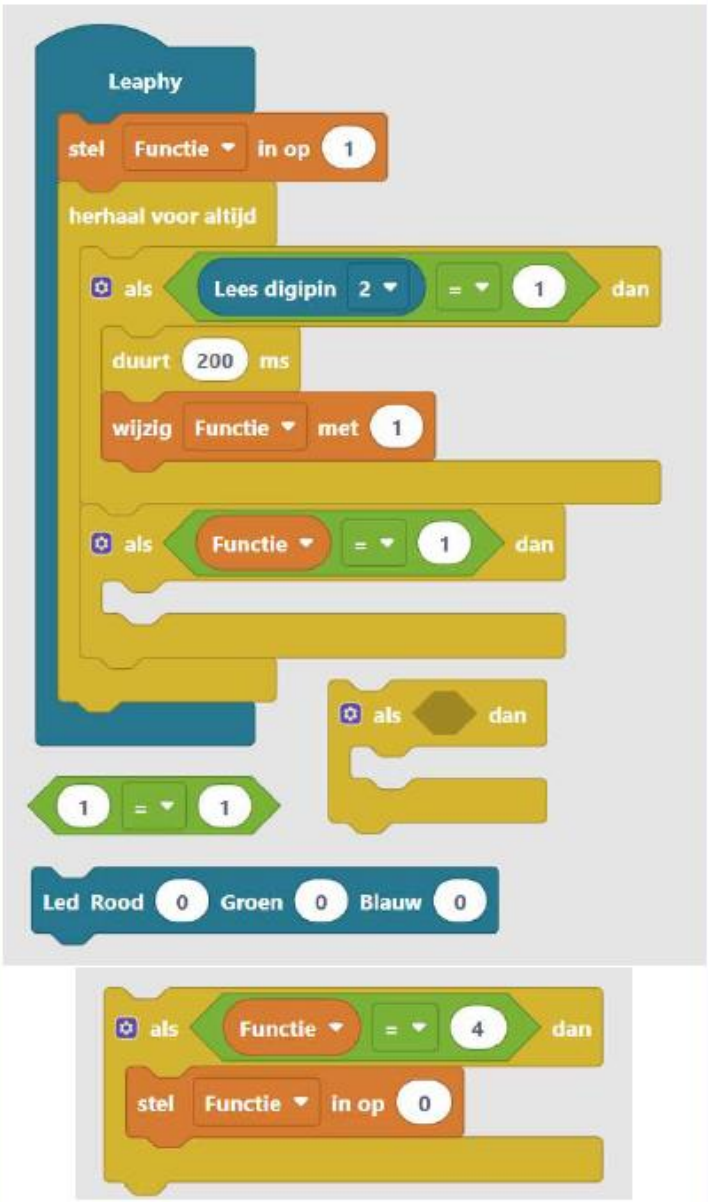
Functie = 0 => led is uit.

Functie = 1 => led is rood.

Functie = 2 => led is groen.

Functie = 3 => led is blauw.

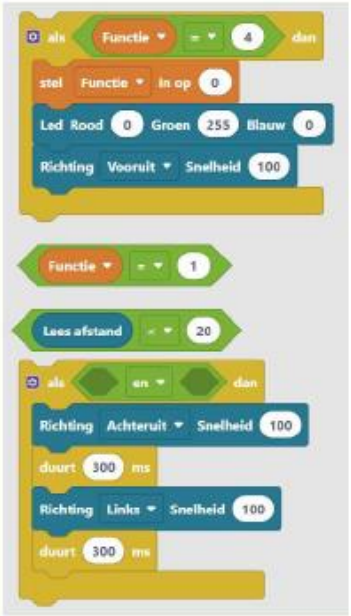
Hiernaast zie je een manier om te zorgen dat de variabele *Fase* niet eindeloos blijft doortellen.



Level 8.9 – Meerdere functies II

Maak een programma waarmee je Leaphy in en uit zijn rijd-en-ontwijk-programma kunt halen.

Let op: als je het gewone Als-Dan-Anders-programma voor de afstandsensor gebruikt, zal de Arduino de drukknop niet uitlezen op het moment dat hij in de afstandsensor-lus zit. Je wilt dat hij ook dan 'oplet'. Dat kun je doen door een dubbele voorwaarde te gebruiken. Hiernaast zie je een manier om dat te doen.



Level 8.10 – Meerdere functies III
Maak een programma waarbij:
Functie = 0 => ledje knipperend in wachtstand.
Functie = 1 => Leaphy rijdt en ontwijkt.
Functie = 2 => Leaphy volgt een lijn.

Maak hiervoor gebruik van je programma bij level 7.8. Hiernaast staat de hoofdstructuur vast klaar.

Je zult merken dat de Arduino soms een beetje onzorgvuldiger wordt in zijn reacties. Je kunt de dender-wachttijd wat aanpassen, of de volgorde van je programma, om het beter te krijgen. Tegelijkertijd heeft de Arduino als computertje ook zijn grenzen. Dus houd er rekening mee dat hij soms even in de war is... Keep calm and keep trying!