

WEEKENDSCHOOL PROGRAMMEREN – BEGELEIDERINSTRUCTIE - ALGEMEEN

1. INTRODUCTIE

Dit document bevat de begeleiderinstructie voor de lescyclus Programmeren. Dit is de informatie die onafhankelijk is van de les. Er is een aparte begeleiderinstructie voor elk van de specifieke lessen.

2. ONBEKEND MET SCRATCH?

Als je nog nooit met Scratch gewerkt hebt, dan moet je voor de les Scratch leren. Als je al ervaring hebt met programmeren in een andere taal dan zul je in een uurtje wel klaar zijn. Als programmeren nieuw is dan zal het zeker een paar uur kosten.

2.1 HULP VOOR BEGELEIDERS

Als je ergens niet uitkomt, schroom dan niet om voor de les te bellen met Hans de Jong, 06 204 234 84. Kan in het algemeen elke dag tot 23 uur. Op de dag zelf kun je nog wel wat vragen, maar er is heel weinig tijd en daarom is het verstandig om alle vragen tevoren beantwoord te krijgen.

2.2 SCRATCH LEREN

Begin hier: <https://scratch.mit.edu/> en kies *Probeer het* of *Bekijk voorbeeld*. Dan programmeer je in Scratch 2.0. In de les programmeren we ook in Scratch 2.0 maar dan op een Raspberry Pi. Daar zijn wat fouten waar we omheen werken, maar verder is het hetzelfde.

Introducties over Scratch vind je hier: <https://scratch.mit.edu/help/>

Je kunt niet precies dezelfde achtergronden, geluiden etc. toevoegen, maar dat zou geen belemmering moeten zijn. Als je dat wel wilt, vraag dan even om de bestanden.

In de les gebruiken we een Raspberry Pi als computer. Die werkt bijna net zoals een Mac of Windows PC. Daar hoeft je je niet op voor te bereiden.

2.2.1 OEFENEN MET EEN RASPBERRY PI?

Mocht je zelf een Raspberry Pi 2B, 3B of 3B+ hebben dan kan ik je ook een image voor een micro SD kaart sturen. Als je wilt kun je ook een complete setup met Raspberry Pi, monitor, toetsenbord etc. lenen.

3. VOOR DE LES

3.1 OPBOUW VAN DE WERKPLEKKEN

We proberen het zo te plannen dat er per begeleider maximaal 8 leerlingen zijn. Die zitten op 4 werkplekken. We proberen meer begeleiders te hebben. Elke begeleider bouwt dus maximaal 4 werkplekken op. Er is een aparte handleiding hoe dat moet.

3.2 KLAAR ZETTEN VAN DE RASPBERRY PI

Als de werkplek klaar is, start dan de Raspberry Pi op. Ze starten zodra je de netvoeding aansluit en die spanning heeft. Als ze allemaal lopen, ga dan pas de tijd instellen op de eerste.

3.2.1 ZET DATUM EN TIJD

Het zetten van datum en tijd gebeurt op één Raspberry Pi. Daarna kun je met een USB stick de tijd overzetten naar alle andere Raspberry Pi's.

1. Ga naar de map *Begeleider*.
2. Dubbelklik *Zet Nieuwe Datum en Tijd.sct*.
3. Voer de nieuwe tijd in. Of geef Enter als de tijd al goed staat.
4. Steek nu de USB stick erin waarop een blauw label *Tijd* zit. Doe dit binnen een minuut.
5. Loop dan de andere Raspberry Pi's af en
 - a. steek steeds de USB stick erin
 - b. wacht tot er een melding rechtsboven komt dat
 - i. de tijd is overgenomen
 - ii. wat de naam van het systeem is en
 - iii. dat je de USB stick kunt verwijderen.



Uitleg: de tijd wordt op de USB stick geschreven als je binnen ca. 1 minuut de USB stick erin steekt nadat je de tijd hebt ingesteld. Als je daarna de USB stick in een andere Raspberry Pi steekt, dan wordt gedetecteerd dat er een nieuwe tijd moet worden gezet. De tijd die gezet wordt is 20 seconden later dan wat op de USB stick staat en de nieuwe tijd wordt teruggeschreven naar de USB stick. Dan wordt de USB stick unmounted en wordt de melding gegeven.

Op deze manier kun je snel en makkelijk de tijd overzetten naar alle RPi's en er wordt rekening gehouden met 20 seconden looptijd tussen de verschillende RPi's.

3.2.2 ZETTEN VAN DE SYSTEEMNAAM

Om straks te zorgen dat we aan het eind van de les het werk van de leerlingen kunnen overnemen op USB sticks moet de naam van het systeem gezet zijn. Als het goed is heb ik dat al gedaan. Als je de tijd instelt of via de USB stick overneemt dan krijg je een rechtsboven op het scherm een melding van de systeemnaam. Controleer of de naam klopt met de naam boven de HDMI poort op de RPi.

Als die niet goed staat, doe dan dit:

1. Ga naar de map *Begeleider*
2. Start *Zet Systeemnaam.sct* op
3. Vul de systeemnaam in.
4. Druk op Enter



4. CLEANUP-SCRATCHFILES

Als de RPi wordt opgestart dan loopt op de achtergrond een proces *CleanUp-ScratchFiles.bash*

Dat proces blijft lopen. Mocht het struikelen of worden gestopt, dan wordt het automatisch binnen een minuut weer opnieuw gestart.

Dit proces behandelt wat eigenaardigheden / fouten in Scratch en geeft waarschuwingen als er iets fout dreigt te gaan. Hier staat wat het doet.

- Als je een .sb2 of .sbx file opslaat in */home/pi* wordt deze weer op het bureaublad gezet als .sb2 file, met de eventuele spaties verwijderd uit de filenaam. Dit is nodig wegens fouten in Scratch 2 op Raspberry Pi. Je kunt hierdoor Scratch files gewoon opslaan in de map die wordt voorgesteld door Scratch 2. Dit is altijd */home/pi*.
- Als er al een bestand met die naam op het bureaublad stond, dan wordt het oude bestand verhuisd naar de map *ScratchPgms-Backup* op het bureaublad. Daar vind je dus alle oude opgeslagen bestanden, mits het opslaan gebeurt in */home/pi* zoals voorgesteld en niet een andere map wordt gekozen.
- Als er een Scratch programma open is en er is gedurende 7 minuten geen Scratch file opgeslagen geweest in */home/pi* dan wordt een waarschuwing gegeven. De waarschuwing blijft ook nog een paar minuten staan nadat je hebt opgeslagen. Je kunt erop klikken om het te verwijderen.
- Na 10 minuten wordt die waarschuwing heftiger: knipperend rood.
- Als er meer dan één Scratch file open is dan wordt er een waarschuwing gegeven. In principe zou het voor les 1B wel mogen, maar omdat het openen langzaam gaat, klikken ze soms nogmaals hetzelfde bestand aan. Dan zijn er twee open en als ze dan in de ene werken en daarna de andere opslaan, dan zijn ze de wijzigingen kwijt (en moeten ze de backup terug halen). Daarom testen we erop. Later moet toegevoegd worden om alleen te waarschuwen als tweemaal hetzelfde bestand open is.



4.1 SPECIAAL VOOR LES 2

- Als het Scratch programma gebruik maakt van extensies voor *scratchClient*, dan wordt een melding gegeven als *scratchClient* niet loopt.
- Als *scratchClient* loopt en er is meer dan één Scratch programma open dan komt er een waarschuwing, want dan werkt *scratchClient* niet.

4.2 HANDIG VOOR DE BEGELEIDERS

- Als er een USB stick met een tijdsbestand wordt aangesloten, dan wordt de tijd gezet op 20 seconde later dan op de USB stick en het tijdsbestand wordt bijgewerkt naar de huidige tijd. Zie boven.
- Als er een USB stick wordt aangesloten waarop het lesmateriaal van de leerlingen moet worden gedownload en een volgende les worden klaar gezet, dan wordt dat automatisch uitgevoerd. Er komt een melding als je de USB stick kunt verwijderen. Optioneel (gespecificeerd op de USB stick) wordt de RPi gestopt.

5. GEZOND COMPUTEREN

- We geven een introductie over hoe ze moeten computeren zonder gezondheidsklachten te krijgen. Dit kan gebeuren voor les 1B op de eerste lesdag of aan het begin van de tweede lesdag. Dit hangt ervan af of we de werkplekken voorzien van de verlaagde tafelbladen.
- Daarna spelen we stukjes video af
<http://www.hetklokhuis.nl/tv-uitzending/1142/Gezond%20computeren>
Van 4:50 tot 8:35 en daarna van 11:07 tot 12:23.
- De volgende lessen vragen we ze om de Klokhuis Gezond Computeren Top 5 te herinneren:
 1. Zet je beeldscherm op armlengte afstand.
 2. Zorg voor een goede ondersteuning van je onderarmen.
 3. Als je alleen de muis gebruikt, geef hem dan alle ruimte.
 4. Probeer links en rechts muizen af te wisselen.
 5. Blijf niet te lang in één houding zitten, maar wissel af. En zorg voor voldoende beweging tussen het computeren door.

6. WELK LEERPROCES?

De Lifelong Kindergarten group van het MIT in Boston, waar Scratch ontwikkeld is, propageert Creative Learning: leren door te doen. Zij leggen de nadruk op het door de leerlingen zelf laten ontdekken en spelen en zo te leren. Ik geloof daar ook zeer in, echter we hebben maar drie zondagen en willen ze toch wat meer meegeven. Daarom is gekozen om ze opdrachten te laten maken. Voor spelen en eigen inbreng is gelegenheid aan het eind. Maar als ze eerder zelf dingen willen uitproberen en een wat andere richting in gaan dan is dat prima, maar kijk wel dat ze niet in een onderwerp blijven hangen omdat ze de rest (zogenaamd) niet snappen.

7. DEELOPLOSSINGEN

Soms hebben de leerlingen hun werk niet goed opgeslagen en zijn ze plotseling heel veel kwijt. In de map `/home/pi/Weekendschool-Programming` vind je per les een map *Tussentapen* waarin per opdracht de status van het programma tot dan staat. Je kunt dan eventueel dat kopiëren en ze daarmee verder laten gaan.

8. ALGEMENE AANWIJZINGEN

8.1 KOPIËREN TUSSEN DE SPRITES

Je kunt blokken kopiëren van de ene naar de andere sprite door de opdracht of het blok opdrachten op te pakken en naar de betreffende sprite te slepen. Je zult zien dat er een lijntje om de sprite komt als die geselecteerd is. Het is het handigste om het te kopiëren blok of de stapel blokken links boven op te pakken.

Je kunt geen opdrachten naar een nieuwe sprite *verplaatsen*. Je moet kopiëren en dan de originele opdrachten verwijderen. Als je niet helemaal zeker bent dat een blok is gekopieerd, klik dan eerst even op de bestemmings sprite om te zien of het echt gelukt is alvorens het oude blok te verwijderen.

8.2 KOPIËREN UIT EEN ANDER SCRATCH PROJECT

Open het andere Scratch project. Exporteer de gewenste sprite, geluid of kostuum. Sluit het Scratch project. Open het Scratch project waar het naar toe moet. Importeer de sprite, het geluid of het kostuum. Als het om de blokken in een sprite ging, kopieer dan de blokken naar de sprite waar je ze hebben wilt en gooi dan de geïmporteerde sprite weg.

8.3 ANDERE SPRITES GEBRUIKEN

Vaak willen ze eigen sprites kiezen. Dat mag. Maar let erop dat ze eerste alle scripts, uiterlijken en geluiden (voor zover van toepassing) hebben gekopieerd naar de nieuwe sprite. Als ze de oude sprite weggooien zijn die kwijt.

8.4 OPSLAAN

Laat ze regelmatig hun werk opslaan. Anders zijn ze het kwijt bij stroomuitval.

Scratch 2.0 op Raspberry Pi heeft een aantal fouten. Een daarvan is dat als je iets wilt opslaan het altijd per default in de */home/pi* map gebeurt. Scratch vergeet waar vandaan het bestand geopend was. Ook is er een probleem als de bestandsnaam spaties heeft. Als je dan het dan weer wilt opstarten door te dubbelklikken dan gaat dat fout. Tot slot, als extension blocks worden gebruikt in les 2, wordt het programma altijd opgeslagen met *.sbx* extensie waar Scratch daarna bij opstarten niets mee kan.

Zoals eerder beschreven loopt er een proces *CleanUp-ScratchFiles.bash* op de achtergrond die dit oplost.

8.5 VOORTGANG NOTEREN

Nu we deze les nog maar een paar keer gedaan hebben en steeds ook iets hebben veranderd willen we uitzoeken hoeveel tijd ze nodig hebben om de les te doen. Zodat we in latere lessen bijvoorbeeld weten dat ze na een uur bij opdracht X moeten zijn. Daarom krijgt elke begeleider een formulier voor de groepjes die hij of zij begeleidt en vult daar op gezette tijden (bijvoorbeeld elk kwartier) in bij

welke opdrachtnummer elk groepje is. Als de leerlingen dat zien, leg ze dan uit dat het geen wedstrijd is, dus dat ze gewoon moeten doorwerken, en dat het er niet om gaat wie het snelste is. We noteren geen namen van de leerlingen. Het is puur voor ons eigen inzicht in het lesverloop.

Of we dat ook echt doen bepalen we in de voorbespreking van de les.

8.6 WAAR ZIJN WE GEBLEVEN?

De volgende lesdag zijn ze vast vergeten tot waar ze gekomen zijn. Daarom staat er een spreadsheet op het bureaublad (*Wie en waar.ods*) waar ze aan het eind van de lesdag in moeten noteren tot welke opdracht ze gekomen zijn.

8.7 WELKE RPI IS VAN WIE?

Gedurende de les gaat iemand met briefjes en plakband rond om de namen van de leerlingen op een briefje op de RPi's te zetten, zodat ze volgende week weer verder kunnen met de juiste RPi.

8.8 TOETSENBORD PAST ZICHZELF AAN

Soms valt om onverklaarbare redenen het toetsenbord terug op Engels UK. Dubbelklikken op dit icoon in de Begeleiders map zet het weer terug op Engels US.



9. BOEKJES MEE NAAR HUIS?

Als we beginnen met les 1A en dus de volgende week doorgaan met de tweede helft van 1B dan worden alle boekjes ingenomen, zodat ze die volgende week weer kunnen gebruiken.

Het is niet de bedoeling dat ze in de boekjes schrijven.

Aan het eind van les 1B mogen ze als ze graag willen het boekje mee naar huis nemen. Maar als ze dat niet heel graag willen, neem ze dan in voor een eventueel volgend gebruik bij een volgende locatie.

Hetzelfde geldt voor les 2A.

10. THUIS VERDER

Kunnen ze thuis verder met wat ze gemaakt hebben?

- Ja, er is een beschrijving hoe ze dat moeten doen. Als ze hun naam invullen in de spreadsheet op het bureaublad (*Wie en waar.ods*) en aangeven dat ze het willen hebben, dan krijgen ze hun werk en een blad met uitleg per email toegestuurd.
- Bij les 2A hebben ze gebruik gemaakt van extra hardware (plankje met eend). Die hebben ze thuis niet, maar ze kunnen het makkelijk aanpassen naar alleen afbeeldingen op het scherm en het toetsenbord in plaats van de knopjes.
- Hebben ze geen computer waar ze thuis op kunnen en/of mogen werken? Laat ze het noteren. We vinden zeker een oplossing.

11. DEBUGSTRATEGIE

11.1 ALGEMEEN

Als je als begeleider door een leerling wordt gevraagd waarom het programma niet doet wat het zou moeten doen dan is het niet altijd simpel om te vinden wat het programma doet en waar de fout zit. Hier volgen wat aanwijzingen hoe te werk te gaan. En dit is voor de leerlingen ook een goede strategie voor het debuggen.

Laat de leerlingen vooral aan jou en aan elkaar uitleggen wat er moet gebeuren en wat ze denken dat er gebeurt.

1. Eén stapel per keer.

Klik eerst op *stop* (de rode achthoek naast de groene vlag) om alles te stoppen en daarna op een enkele stapel van blokken. Breidt dit geleidelijk uit.

- Soms zit er code in andere sprites die ook wordt uitgevoerd bij het drukken op de groene vlag of andere events. Als je een enkele stapel neemt dan kijk je alleen naar die enkele stapel
- Laat ze de stapel blokken in een paar delen uit elkaar trekken als het te groot is.

2. Kijk naar de waarden die ze invullen. Ze willen nog wel eens spaties of andere tekens ervoor of erna invullen. En dan werkt het niet.

3. Gebruik *zeg* of *denk* blokken om het programma de waarde van een variabele te laten zien (zoals printf in C of ander programmeertalen)

4. Bij verschillende opdrachten zijn er varianten die een actie starten en dan doorlopen en varianten die de actie starten en dan wachten totdat die klaar is. Het is zeer essentieel dat de goede variant gekozen wordt.



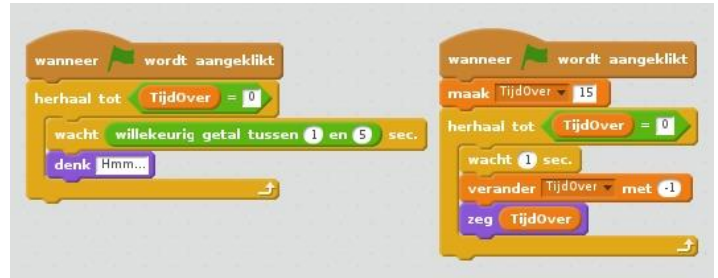
5. Goede plaatsing van stop opdrachten

- Soms zien we dat de opdracht om alles te stoppen op het verkeerde niveau in de blokken staat. Ze klikken dan op de groene vlag en na een paar seconden stopt alles. Vaak is dat doordat de stop opdracht verkeerd geplaatst is.
- Soms komt de stop opdracht te vroeg. Bijvoorbeeld als er geluid wordt afgespeeld en in het midden daarvan wordt een stop opdracht uitgevoerd, dan stopt ook het geluid. De twee blokken hiernaast laten een in de kiem gesmoorde miauw horen. Dat is anders dan wanneer een stapel blokken het einde heeft bereikt.



6. Let op race condities

- Als je deze twee stapels blokken hebt, eventueel in verschillende sprites, dan is het onbepaald wat het resultaat is.
 - Het kan zijn dat eerst de linker stapel gaat lopen en onmiddellijk stopt omdat *TijdOver* nog steeds nul is.
 - Het kan ook zijn dat de rechter stapel gaat lopen en wel eerst de variabele op 15 zet. En dan zal de linker stapel wel gewoon 15 seconden gaan lopen.
- Het lijkt dat links geplaatste stapels eerst beginnen. Maar dat is geen goede strategie. Beter is het in zo'n geval om de linker stapel te laten wachten tot *TijdOver* ongelijk aan 0 is. Of stuur een bericht uit de rechter stapel en laat de linker stapel beginnen als dat bericht wordt ontvangen i.p.v. als de groene vlag wordt aangeklikt.



7. Dubbel geopende Scratch bestanden

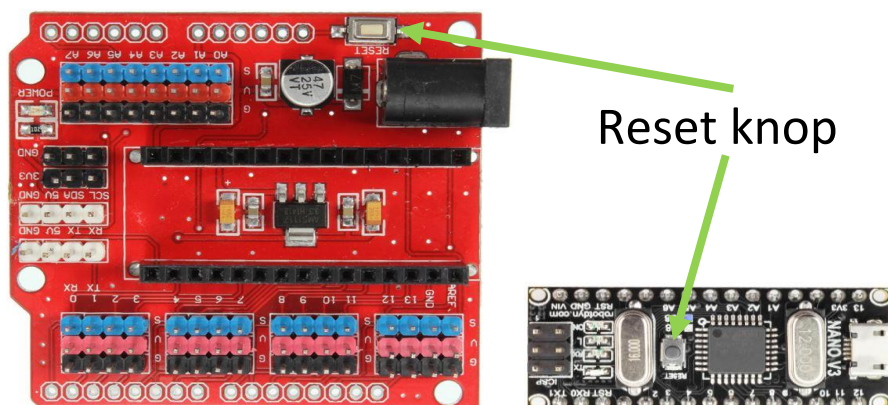
Soms is het opstarten van Scratch wat langzaam. Zoals eerder genoemd dubbelklikken ze dan nogmaals en het resultaat is dat er twee instantiaties open zijn. Ze maken aanpassingen in de een en slaan de ander op en zijn dan hun werk kwijt. Wees beducht op dubbele open Scratch files. Er wordt ook voor gewaarschuwd.

11.2 SPECIAAL VOOR LES 2

8. Is er verbinding tussen Scratch en de Arduino?

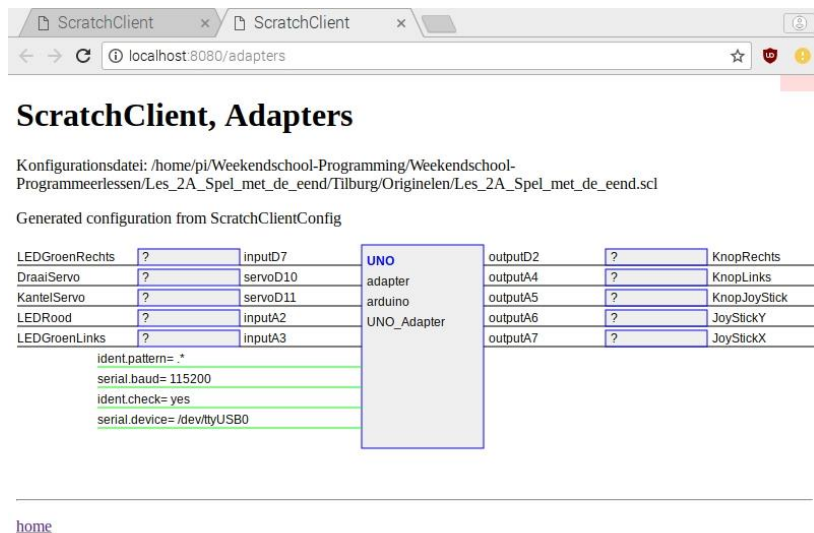
Controleer of de blauwe LED op de Arduino langzaam knippert. Alleen dan is er verbinding tussen Scratch en de Arduino, via *scratchClient*. Als dat niet zo is controleer dan:

- Is de LED uit? Druk dan op de Reset knop van de Arduino of van het adapterbord. De LED moet nu snel gaan knipperen. Als de ene knop het niet doet, neem dan de andere. Sommige knoppen zijn defect op sommige van de plankjes, maar minstens één van de twee werkt wel.



- Is *scratchClient* gestart en loopt die nog steeds? Zo niet, start die opnieuw door te dubbelklikken. Let op: het icoon op de taakbalk voor *scratchClient* verdwijnt zodra *scratchClient* loopt, zodat de leerlingen het niet kunnen sluiten.





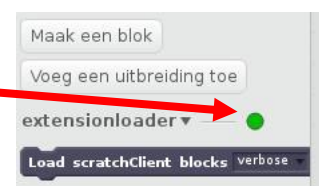
Hier kun je zien of scratchClient loopt. Dit moet groen zijn. Elke keer dat scratchClient opnieuw wordt gestart wordt er in Google Chrome ook een nieuwe tab geopend. Let erop dat je naar de nieuwste – meest rechtse – tab kijkt. De andere tabs kun je sluiten. Het venster van Chrome wordt ca. 10 seconde na het starten geminimaliseerd.

Als alle tabs in Google Chrome gesloten zijn, dan kun je een nieuwe tab openen door dit icoon te dubbelklikken. Het zit in de map Begeleiders.



Bedenk dat scratchClient niet de status bijhoudt. Het wordt naar de browser gestuurd als er een waarde verandert. Daarom zul je na het starten van Chrome vraagtekens zien totdat er een waarde veranderd wordt, ook als er in het verleden al een waarde gestuurd is.

- Zit de USB kabel tussen de Arduino en de Raspberry Pi er in?
- Zijn er meer Scratch programma's open? Er mag slechts één programma open zijn.
- Hebben ze in Scratch wel gedrukt op de sprite met de groene punt? Dat moet elke keer na opnieuw openen van het Scratch programma. Probeer het nogmaals te doen.
- Zitten er nog wel de originele blokken in de sprite met de groene punt? Dit moet erin zitten. De blokken staan helemaal rechts in het venster om weggooien te ontmoedigen. Soms gooien ze die toch weg. Dan kun je het er opnieuw in zetten.
- Zijn de extensies nog steeds actief? Je moet een groene stip zien naast *extensionloader*. Druk op de groene punt bij de start vlag als de stip rood is.
- Meestal lost het bovenstaande het op. Als het dan nog steeds niet wil, trek dan de USB kabel eruit en sluit die opnieuw aan. Drukken op de groene sprite lijkt niet niet nodig, maar het kan ook geen kwaad.



9. Kijk naar de waarde van de variabelen in de browser.
Als je op een knop drukt, zie je dan de waarde veranderen? Als je een waarde naar de Arduino stuurt, is die dan zichtbaar in de browser?

outputD2	1	KnopRechts
outputA4	1	KnopLinks
outputA5	1	KnopJoyStick
outputA6	532	JoyStickY
outputA7	527	JoyStickX

10. Wacht tussen servo bewegingen

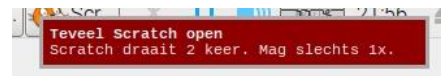
- Een servo heeft tijd nodig om naar de juiste positie te draaien. De eend zal met de linker blokken



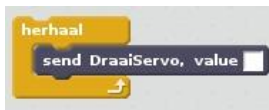
geen buiging maken, want de buiging naar 90 graden wordt direct teniet gedaan door het draaien naar 0 graden. Als je 0,2 seconde pauze neemt dan werkt het wel, zie rechts.

11. Kijk naar foutmeldingen op het scherm.

- Rechtsboven op het scherm komen meldingen als er fouten worden gevonden.

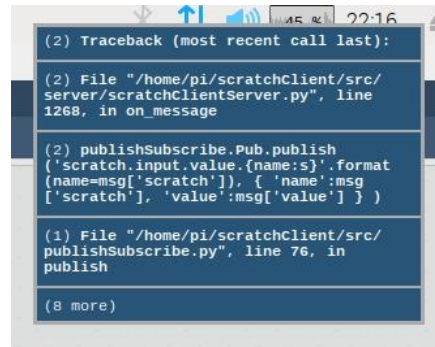


12. Als je rechtsboven foutmeldingen krijgt zoals hier, dan komt dat mogelijk doordat je een vaak een send blok uitvoert waar geen waarde is ingevuld, bijvoorbeeld zo. Dit is een fout van scratchClient. De



enige remedie is momenteel om scratchClient opnieuw te starten. De oude wordt

vanzelf afgesloten. Wel in Scratch weer op de groene sprite in het speelveld klikken (naast de startvlag).

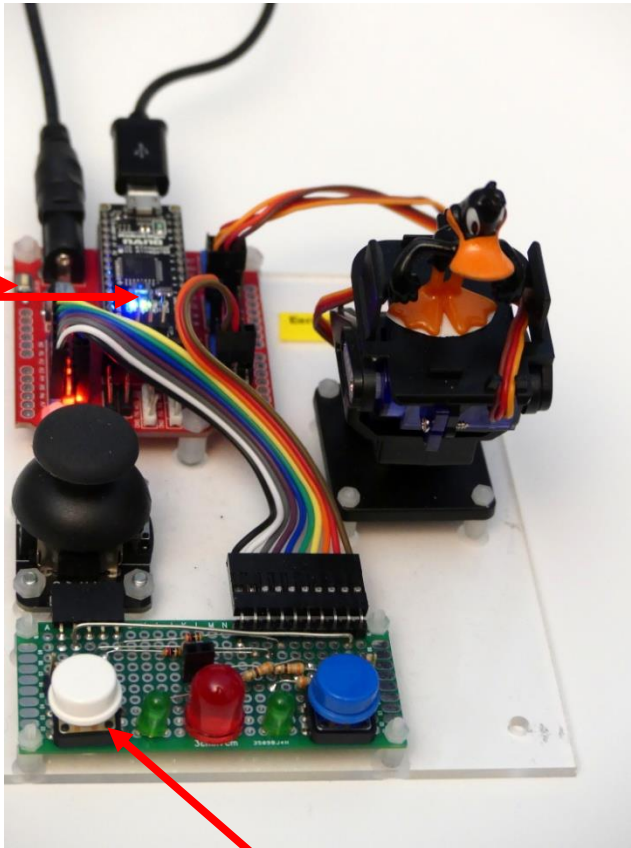


13. Zorg dat de 9 volt adapter is aangesloten

Soms kan het plankje functioneren zonder dat de 9 V voeding is aangesloten. Maar als bijvoorbeeld twee servo's tegelijk bediend worden kan het zijn dat de Arduino (en soms de RPi) resetten.

14. Als je het idee hebt dat er iets kapot is in de hardware van het plankje, dan kun je een Power On Self Test (POST) laten doen.

Stap 2:
Reset



- Stap 1: Druk de linker knop in en **houd die ingedrukt**
- Stap 2: Druk op de reset knop op het expansiebord of op de Arduino en laat weer los.
- Stap 3: Wacht tot er lampjes gaan branden en de eend gaat bewegen.
- Stap 4: Laat dan pas de linker knop los.
- Nu kun je testen:
- Stap 5: De eend moet naar links en naar rechts draaien en buigen in het midden. Dat duurt ca 10 seconde. Daarna kijkt de eend naar beneden in de middenstand.
- Stap 6: Kijk naar het linker groene LEDje. Die moet knipperen.
- Stap 7: Druk de linker knop in. Het linker LEDje moet nu constant branden.
- Stap 8: Kijk naar het rechter groene LEDje. Die moet knipperen.
- Stap 9: Druk de rechter knop in. Het rechter LEDje moet nu constant branden.
- Stap 10: Kijk naar de grote rode LED. Die moet knipperen.
- Stap 11: Druk de joystickknop in. De rode LED moet nu constant branden.
- Stap 12: Als de eend klaar is met bewegen, dan kun je met de joystick de eend bewegen. Dit kan naar links/rechts en naar onder/boven.
- Stap 13: Plak de eend weer op het platform als die er door stap 12 af geslagen is 😊.

11.3 TOT SLOT

Debuggen is een creatief proces. En je kunt je blindstaren op een bepaalde plek. Iemand anders laten meekijken kan heel zinvol zijn. Maar vooral moeten de leerlingen je kunnen volgen. Het beste is als je zelf ziet wat er aan de hand is en dan vraagt hoe iets werkt, laat ze de stapel blokken uitleggen. En zie

erop toe dat ze eerst nadenken, het jou uitleggen en dan de verandering maken. Vaak slepen ze in het wilde weg blokken van de ene naar de andere plek zonder te begrijpen wat ze doen. En begrijpen wat ze doen is uiteraard de essentie van de les.

12. AFSLUITEN, INPAKKEN EN OPRUIMEN

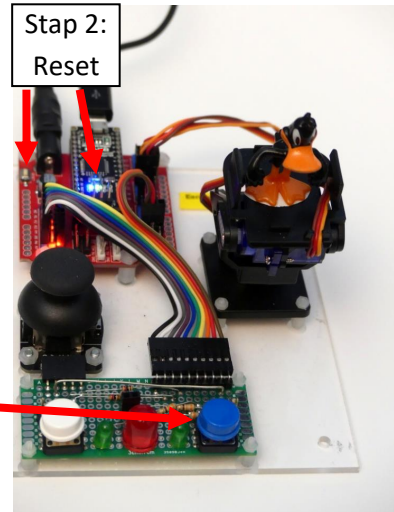
Bij het eind van de dag moet dit gebeuren.

12.1 ACTIES VAN DE LEERLINGEN

1. Vraag de leerlingen of ze de spreadsheet hebben ingevuld met hun naam en waar ze gebleven zijn. Zo niet, laat ze dat dan alsnog doen.
2. Als we de les hier afsluiten vraag dan ook of ze hebben ingevuld of ze thuis verder willen. Ook als ze thuis geen computer, tablet of smartphone hebben. We vinden een oplossing. Om thuis verder te gaan met les 2B is het plankje **NIET** nodig, dus dat is geen argument.
3. Laat ze de spreadsheet opslaan en afsluiten.
4. Laat ze het Scratch programma opslaan en afsluiten.

12.2 AFSLUITEN VOORBEREIDEN

5. Controleer zelf dat er geen programma's meer open staan. Sla op en sluit af wat toch nog open staat.
6. Zet de eend in inpakstand voordat je die afkoppelt van USB / 9 volt voeding (een van de twee is nog nodig).
Stap 1: Druk op de rechter knop **en blijf die indrukken**.
Stap 2: Druk op de resetknop op het expansiebord of op de Arduino Nano en laat weer los.
Stap 3: Wacht tot de eend gaat bewegen.
Stap 4: Laat de rechter knop los.
Stap 5: Wacht tot de eend stopt met bewegen.
Stap 6: Koppel USB en 9 volt voeding af.



12.3 WEL OF NIET AFSLUITEN?

1. Als we volgende week doorgaan met dezelfde les, sluit dan de Raspberry Pi af via het menu. Als er daarna gedurende 5 seconde geen LEDjes in de Raspberry Pi branden dan kan de stekker eruit.
2. Als we niet doorgaan (b.v. eind van de derde dag), laat de Raspberry Pi dan lopen. Iemand gaat met een USB stick langs alle werkplekken om het werk op te slaan op de USB stick en daarna af te sluiten.

13. VRAGEN EN ANTWOORDEN

Hier is een aantal vragen die ze mogelijk stellen en de antwoorden

- Waarom heet dit ding Raspberry Pi?
 - Het beestje moet een naam hebben.
 - De bedenkers wilden een naam van fruit gebruiken om nostalgische redenen. Raspberry is een framboos. Vroeger werden meer computers naar fruit genoemd: Apricot (abrikoos) en je kent er vast nog wel een: Apple.
 - Pi komt van Python, de naam van een programmeertaal die je kunt gebruiken. En als je Pi spelt als Pie dan betekent het: taart. Dus frambozentaart.
 - Het heeft navolging gekregen, want iemand is Banana Pi gestart.
 - En er is nu ook een Orange Pi, wat een complete computer is ala de Raspberry Pi, maar dan voor rond de 10 euro (excl. SD kaart, voeding, monitor, toetsenbord en muis).
- Wat kost een Raspberry Pi en deze opstelling?
 - Deze Raspberry Pi, als je die in Eindhoven koopt kost (sossolutions.nl) 62,95 euro. Dit omvat de Raspberry Pi, de behuizing, de SD kaart en netvoeding. Gratis verzending.
 - Tweedehands monitoren kun je kopen voor ca. 25 euro, degene die we in de les gebruiken waren tussen de 8,50 en 10 euro. Tweede hands muizen 1,25 euro en toetsenbord 2,50 euro. Kabel voor de monitor kost een paar euro. Maar het hangt heel erg af waar iets toevallig aangeboden wordt en er kunnen verzendkosten bijkomen.
 - Om alles nieuw te kopen moet je naast de Raspberry Pi rekenen op ca. 120 euro (100 euro monitor, 15 euro muis en toetsenbord, 5 euro HDMI kabel).
 - Overigens zijn er ook mogelijkheden om een heel klein scherm aan te sluiten en dan ben je minder kwijt, maar heb je ook minder resolutie.
- Zijn er nog goedkopere computers?
 - Ja. Afgezien van de Orange Pi die ik al wel heb maar nog geen tijd voor gehad om uit te proberen zullen we in de tweede en derde les een computer gebruiken (Arduino Nano) die samen met de voeding en wat cijferdisplays en knopjes ca. 5 euro kost. Maar daar kun je dan ook veel minder mee, en je hebt een grotere computer nodig (zoals de Raspberry Pi) om die te programmeren.