Projecten 2

*De Raspberry Pi gameconsole*

Met behulp van een Raspberry Pi wordt een gameconsole nagebootst om zo oudere videospelletjes te kunnen emuleren.

Van der Velpen Dwight

8/20/2017

Projecten 2

De Raspberry Pi gameconsole

Inhoud

[Inleiding 2](#_Toc491020564)

[1. Eindrapportering 3](#_Toc491020565)

[a. Eindresultaat 3](#_Toc491020566)

[2. Gebruikte onderdelen met uitleg 4](#_Toc491020567)

[a. De Raspberry Pi 4](#_Toc491020568)

[b. Arduino 6](#_Toc491020569)

[c. Lakka 7](#_Toc491020570)

[3. Werkwijze en problemen 9](#_Toc491020571)

[4. Zelfevaluatie 11](#_Toc491020572)

[Inhoud van het portfolio 13](#_Toc491020573)

[Bronnen 14](#_Toc491020574)

[Bijlagen: 15](#_Toc491020575)

# Inleiding

Projecten 2 is een studieonderdeel waar een student zichzelf een eigen gekozen project oplegt en deze stapsgewijs probeert te realiseren. Zo heb ik zelf gekozen om met behulp van een Raspberry Pi een console te maken die het mogelijk maakt oudere videospelletjes te emuleren, en dan vooral spelletjes die afkomstig zijn van de jaren ’80 en ‘90. Deze zijn hedendaags onmogelijk duur om te bemachtigen en zijn daarbovenop meestal zeer schaars. Nochtans zullen vele mensen die videospelletjes verzamelen, spelen en ontwerpen het zeer belangrijk vinden dat mensen deze ook zouden kunnen spelen. Deze spelletjes, zoals Pacman, Pong, Asteroids… zijn de klassiekers van deze media-vorm.

Als kind had ik zelf een Sega mastersystem en een Famicon, 2 oude klassieke consoles waarvan ik vele computerspelletjes van had verzameld. Daarna kreeg ik een Nintendo 64, i.p.v. andere consoles die in die tijd veel populairder waren. Zo is er bij mezelf van zeer jong een sterke emotionele band opgericht voor deze oude spelletjes. Door de jaren heen zijn deze consoles veel versleten. Sommigen van deze consoles werken niet meer of zijn in een staat waar deze amper functioneren. Dit zorgt ervoor dat deze spelletjes niet speelbaar zijn behalve als ik enkele honderden euro’s zou willen spenderen om deze consoles te vervangen.



Het idee van dit project is niet om hedendaagse spelletjes te kunnen emuleren om zo piraterij te promoten. Echter is het een idee om via een Raspberry Pi een enkele console te maken die in staat is om diverse oudere consoles te emuleren. Deze games zijn ondertussen al enkele tientallen jaren oud en zouden, eerlijk gezegd, geen honderden euro’s moeten kosten om te ervaren. In dit project zullen er enkel games geëmuleerd worden die ik zelf in bezit heb in een legale vorm.

Het originele idee was om de vooruitgang in een blog bij te houden en zo langzaam het project op te bouwen maar dit is uiteindelijk vervangen geweest door een handleiding om deze console te repliceren op een eenvoudige manier. De blog werkte niet zozeer wegens dat ikzelf nogal sporadisch werkte aan dit project. Zo werd er sommige weken amper vooruitgang geboekt en werd er eerder onderzoek gedaan, terwijl er op andere momenten veel praktisch werk werd verricht en zo een hele vooruitgang geboekt werd.

In dit verslag tracht ik een beeld te schetsen van hoe het project tot stand kwam en welke stappen ik hiervoor ondernomen heb.

# Eindrapportering

## Eindresultaat

Het doel van het project kan het best omschreven worden met de volgende opstelling:

* Onderzoeken Raspbery Pi. -Voltooid
* Instaleren en configureren Raspberry Pi. -Voltooid
* Onderzoeken OS -Voltooid
* OS gebruiken en configureren. -Voltooid
* Onderozken Controller schema's. -Voltooid
* Fabriceren Breadbord Prototype. -Partieel voltooid
  + Bekabeld -Voltooid
  + Bluetooth -Partieel voltooid
  + RC -Niet voltooid
* Fabriceren PCB prototypes. -Pas mogelijk na Breadboard
* Behuizen PCB -Pas mogelijk na PCB

Er is sprake van een werkende “console” met controller. Echter is er een probleem opgetreden met de bluetooth die deels opgelost is met een 2e module. Er zou een uitbreiding gemaakt kunnen worden om met een andere bluetooth module te werken die wel van nature compatibel zou zijn, maar dit vergt iets meer werk.

Het radio gedeelte is hierdoor niet meer opgenomen. Het doel om een controller op 3 diverse manieren te gebruiken was onnodig complex en nuttig te verwezenlijken. De huidige stand laat toe om externe controllers in te pluggen, zoals een Xbox controller, een zelfgemaakte controller en een toetsenbord. Deze zijn ruimschoots voldoende.

Het project was meer werk dan verwacht was. Er werd gekozen voor een project met hoge einddoelen die uiteindelijk verlaagd zijn moeten worden. Er was veel beter geopteerd voor een ietwat simpelere doelstelling, zoals enkel het emuleren van games via Raspberry Pi, en uitgebreid.

Maar dit wil niet zeggen dat het project mislukt is. De essentie van het project is behaald, het emuleren van ouderwetse games op een Raspberry Pi. Er is in de loop van het project onderzoek gedaan rond de Raspberry Pi, over de diverse emulatie OS’en, over hoe hedendaagse controllers opgebouwd zijn, over Human Interface Devices…

Er is veel bijgeleerd en er is een duidelijk werkpunt aan het project om deze nog verder uit te breiden. Het is enkel de controller die iets moeilijker bleek te zijn dan verwacht.

# Gebruikte onderdelen met uitleg

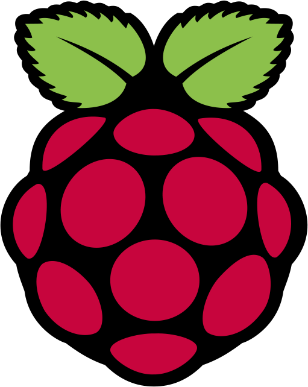
## De Raspberry Pi



Het eerste, en belangrijkste, component van het project is de Raspberry Pi. Raspberry Pi is de naam van diverse singleboardcomputers gebaseerd op een ARM-processor die tegen een minimale prijs worden vervaardigd en verkocht. De Raspberry Pi werd ontwikkeld aan de Universiteit van Cambridge en was bedoeld voor educatieve doeleinden. [1] De Raspberry bestaat al sinds 2012. Sindsdien zijn er verschillende modellen en versies uitgebracht. [2] Elke uitbreiding heeft zo zijn eigen voor- en nadelen tegenover zijn voorgangers. Het gebruikte model in dit project is de Raspberry Pi 3B. Dit is het nieuwste model dat ingebouwde Wi-Fi en Bluetooth modules bevat.

De Raspberry Pi zou sterk lijken op de open structuur van de eerste IBM Pc’s. De GPIO bus maakt het mogelijk om een add-on-board aan een Raspberry Pi te koppelen. De aanwezigheid van [USB](https://nl.wikipedia.org/wiki/Universal_Serial_Bus)-poorten, een [DSI](https://nl.wikipedia.org/w/index.php?title=Display_Serial_Interface&action=edit&redlink=1)- en een [CSI](https://nl.wikipedia.org/w/index.php?title=Camera_Serial_Interface&action=edit&redlink=1)-connector maken het mogelijk om meerdere USB-devices en een display en een camera aan het moederbord te koppelen.

Het is ook zeer belangrijk om te vermelden dat de Raspberry Pi gebruik maakt van een Live-systeem. Dit is een systeemstructuur dat gebruik maakt van live-media, (cd-roms, USB-sticks, SD-kaartjes, …) om het besturing systeem op te bewaren en te laten draaien. Dit heeft natuurlijk als nadeel dat je als eerste afhankelijk bent van de media dat het niet vastloopt of corrupt wordt. Daarnaast is deze media niet ontworpen voor dit soort gebruik. [3] Een standaard SD kaart heeft namelijk een ReadWrite cycli totaal van ongeveer 100.000 cycli. En dit is zonder rekening te houden met het hotspot fenomeen. Dit zorgt ervoor dat de media meerdermaals gelezen en geschreven wordt op dezelfde plek waardoor deze sneller corrupt worden. [4] Dit is ideaal voor projecten wegens deze snel vervangen kan worden en er zo met 1 enkele Raspberry Pi, meerdere projecten uitgevoerd kunnen worden. Er dient echter wel een drager aangekocht worden per project.

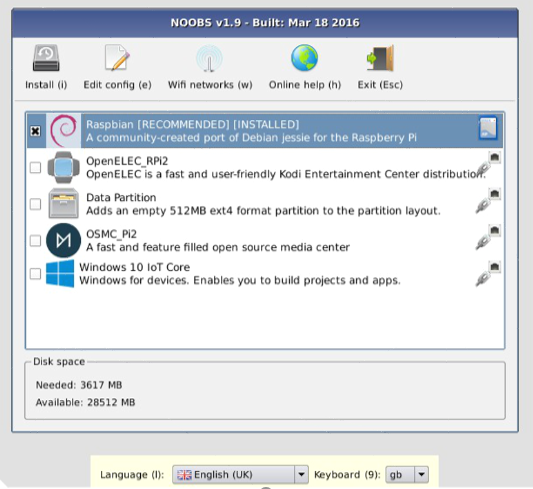
Naast de nadelen die alreeds vernoemd zijn, bestaan er ook veel voordelen van live media. Het eenvoudig verwisselbare systeem zorgt ervoor dat deze eenvoudig te verwisselen is en te back-uppen zonder enige zorg. Er is daarom een aparte handleiding (zie eerste uitbreiding). Het proces is niet al te complex maar veelal is er voor zulke kern doelen geen eenvoudige handleiding paraat, hierdoor kunnen er vele SD kaarten geformatteerd worden wanneer er fouten optreden, en kan men het project opnieuw maken. Met mijn beknopte instructies zal het heel wat eenvoudiger zijn om een SD-kaartje te herstellen en zo opnieuw voor andere dingen te gebruiken of de vooruitgang te back-uppen in geval van nood.

De Raspberry Pi maakt veelal gebruik van NOOBS. Dit is een New Out Of Box Software wat het installatie proces vereenvoudigt. Deze hoeft enkel op de lege SD kaart gekopieerd worden en zal bij opstart een installatie scherm tonen waarbij de verschillende OS’en ter beschikking staan. Deze zijn niet alle mogelijke OS’en die je kan instaleren op de Raspberry maar eerder een “Staff pick” van de meest betrouwbare en geteste software. Het is zeer handig wanneer iemand nog niet bekend is met Raspberry Pi om zelf deze software te gebruiken als een startpunt. Het is een eenvoudige installatiewizard om de gewenste OS’en op het SD kaartje te installeren.

Een van die OS’en is Raspbian. Dit is een Debian versie die speciaal geoptimaliseerd is om te gebruiken op de Raspberry Pi. Voor personen die weinig ervaring hebben met Linux of Raspberry Pi zal deze versie zeer gewaardeerd worden omdat er allerlei tools in verwerkt zitten om met de software te leren werken. Het is een visuele OS, wat een overschakeling van bv. Windows veel vlotter doet verlopen. Zo zijn er ook andere tools die helpen met basis functionaliteit zoals een verkenner voor de filesystemen, een browser…

Enkele andere OS’en die beschikbaar zijn via de NOOBS zijn Lakka, OSMC, OPENELEC\_RPI2, WINDOWS 10 IoT Core, ...

NOOBS is verkrijgbaar op een SD-kaart, waar dit vooraf op geïnstalleerd staat. Online vragen ze daar al snel enkele euro’s voor en ik raad iedereen ten sterkste af om dit te doen. Het draait hier namelijk rond een simpele SD kaart waar er een image op gekopieerd is, iets wat zelf kan gebeuren op minder dan een minuut tijd zonder enige ervaring. Daarbovenop staan er op de SD-kaartjes meestal oudere versie van de Full NOOBS-installatie. Dit wilt zeggen dat er een hoop OS’en ontbreken die tegenwoordig wel ondersteund zijn, er een goedkoop kaartje gebruikt word om deze op te installeren en er een hoop OS’en opstaan die je niet zal gebruiken.

 et is veel beter om de LITE-versie van NOOBS te downloaden (+- 50Mb) in plaats van deze aan te kopen. Deze kan eenvoudig gekopieerd worden op een leeg SD-kaartje van minimum 4Gb. Hiervoor is er wel een internetverbinding nodig maar deze zal in NOOBS via een eenvoudig schermpje bij opstart gevraagd worden. Daarna kan je uit de volledige collectie van OS’en kiezen die online gehost worden. Wegens ik een Raspberry Pi 3B heb, met Wifi antenne, is dit zeker een voordeel.

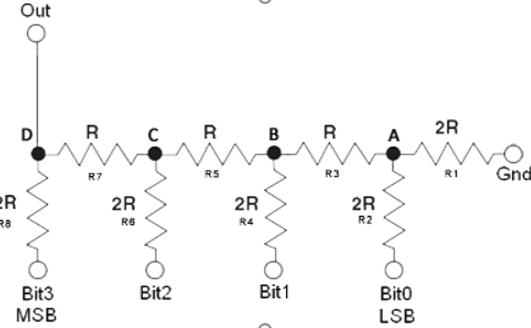
Verder kan er nog vermeld worden dat er best een behuizing voor de Raspberry voorzien wordt, deze zorgt ervoor dat de hardware zelf niet beschadigt. Deze zijn ook verkrijgbaar in bundels maar kan evengoed zelfgemaakt worden. Er zijn echter 4 montage boringen voorzien op de Raspberry Pi waarmee je deze kan bevestigen.

## Arduino

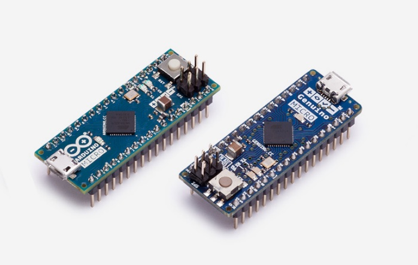
De aanschaf van de Arduino kwam uit een probleem: er werd geen oplossing gevonden om een controller te maken die eenvoudig uit te breiden was en eenvoudig te gebruiken is. Er was een mogelijkheid om de GPIO-pinnen te gebruiken die op de Raspberry Pi al reeds ter beschikking waren. Dit was een eerder complex concept en er zou geen mogelijkheid zijn om de lay-out van de knoppen simpel op te slaan.

Er werd geopteerd om te werken met een Arduino Micro. Deze heeft namelijk HID-mogelijkheden en kan zo gebruikt worden om via eenvoudige drukknoppen een controller te maken. De Arduino zelf word dan verbonden met een USB-kabel met de Raspberry Pi en bevat code om zich te gedragen als een eenvoudig USB-apparaat. Deze zal verbonden worden met enkele drukknoppen.

Deze controller zal dan bestaan uit een tiental drukknoppen, elk verbonden aan een pin van de Arduino. Enkel pinnen D0 en D1 kunnen niet benut worden wegens deze standaard al aan de USB Tx en Rx verbonden zijn waardoor de communicatie verstoord zou kunnen worden. De analoge pinnen kunnen ook gebruikt worden door ze als A0 t.e.m. A5 te benoemen en ze daarna in te lezen met de Digital Read. De analoge pinnen zullen dan gebruikt worden als digitale pinnen. Ze zouden ook via een R2 netwerkje kunnen opgedeeld worden waardoor er meerdere drukknoppen aan dezelfde Analoge pin gekoppeld kunnen worden. Hierbij moet er wel rekening gehouden worden met aflopende correctheid wegens weerstands- onnauwkeurigheden. Er wordt theoretisch vanuit gegaan dat er 8 knoppen kunnen verbonden worden per pin, maar wegens veiligheidsredenen zou ik er persoonlijk maar 4 per pin bevestigen. Dit is in mijn opinie trouwens veel meer dan nodig zou zijn voor een basis controller.



Verder werd er voor de micro gekozen voor zijn lage prijs en gemiddeld aantal pinaansluitingen. Er wordt in dit project gebruik gemaakt van zo goed als elke pin.



De Arduino zelf is een [opensource](https://nl.wikipedia.org/wiki/Open_source)-[computerplatform](https://nl.wikipedia.org/wiki/Computerplatform) bedoeld om [Microcontroller](https://nl.wikipedia.org/wiki/Microcontroller)s eenvoudig te maken. Ze maken gebruik van shields en diverse opensource schema’s waardoor de marktprijs drastisch lager ligt dan andere microcontrollers. Er zijn ondertussen veel modellen beschikbaar, met elk een andere specialiteit. De UNO is zowat het vlagschip voor Arduino, het bevat zowat de meest benodigde connectoren en kan zo gebruikt worden om in diverse testopstellingen geïmplementeerd te worden. Daarnaast hebben we dan de MEGA, die zeer veel pinnen ter beschikking biedt, de Nano, die zeer klein is en daardoor gebruikt kan worden in kleinschalige projecten, de Mini, die nog kleiner is dan de Nano…

De Micro heeft desondanks een kleine afmeting, nog steeds de mogelijkheid op ICSP-pinnen en een micro-USB poort te voorzien.

Verder wordt de software van Arduino geschreven in de IDE zelf. Deze is wederom opensource en heeft daarom enkele alternatieven zoals de voormalige codebender, Stino, Visual studio, en dergelijke. Ook vermeld ik snel even dat er enkele simpel simulators te vinden zijn die het proces op zich ook weer vereenvoudigen. Deze zijn: Simulator For Arduino, 123D circuits, Arduino debugger enz.

## Lakka

De OS die gebruikt word om de videospelletjes op te emuleren is Lakka, echter was deze in het begin van het project Retropie. Retropie is net zoals Lakka een OS voor op Raspberry Pi die een Gui voorziet waarin meerdere emulatie software en ROM’s (games) kunnen gekozen worden. Het is vooral gemaakt om met een controller bestuurbaar te zijn en als de hub te dienen voor de “console”.

Er waren enkele problemen met Retropie die niet opgelost raakten. Er werd beroep gedaan op het officiële forum en het subreddit maar ik merkte al snel op dat vele threads die de problemen die ik voorhad beschrijven, niet werden opgelost. Er werd contact opgenomen met een prominente user in deze fora maar deze bleek weinig hulp te bieden. Retropie was duidelijk geen OS waar dat een actieve nuttige userbase mee verbonden was.

Lakka kwam ter vervanging van Retropie. Lakka en Retropie zijn beide afgeleid van de opensource RetroArch en de libRetro structuren. Dit wilt zeggen dat ze beide gebruik maken van dezelfde emulatie software. Lakka is veel gebruiksvriendelijker en veel makkelijker en vanzelfsprekend in aanpassing voor het project. Er werd wat gepuzzeld om alles in elkaar te passen en zo kwam ik uiteindelijk op een werkende versie van Lakka met enkele kleine aanpassingen in de geluidsinstellingen. Het probleem hier is namelijk dat de Raspberry Pi niet automatisch een audiokanaal kiest. Er is namelijk een stereoplug, een HDMI poort, meerdere USB poorten en de GPIO pinnen. Dit is opgelost door een script te laten uitvoeren voordat Lakka opstart. Hierbij zal er een audiokanaal gekozen worden, en zal er later ook hotswapping mogelijk gemaakt worden.

Lakka biedt zo ook al reeds ingebouwde, en geteste, emulatie software aan. Deze ontbrak bij Retropie en hierdoor was Lakka al een stap gebruiksvriendelijker. Lakka had echter wel enkele nadelen. Het is een minder gekend OS. Hierdoor was de community iets kleiner en was er minder materiaal om mee te werken. Er waren geen gedetailleerde beschrijvingen van reeds voorgaande projecten, waar dit project op gebaseerd werd, dus moesten vele delen zelf uitgezocht worden.

Lakka is ondertussen wel, in tegenstelling van Retropie, ondersteund door NOOBS.

Enkele uitbreidingen die zelf onderzocht werden in Lakka zijn:

* De mogelijkheid om bij opstart een andere OS te selecteren om debugging en file management mogelijk te maken. (Raspbian)
* Een audio-selecteur.
* Hotswapping van een USB stick met hierop de ROM’s.
* Automatische updater van de ROM bibliotheek in Lakka zelf.

# Werkwijze en problemen

In het eerste deel van het project, van het begin tot tussenrapportering 1, werd er een zeer goede vooruitgang geboekt. Er werd een werkende Lakka build geïmplementeerd op een Raspberry Pi die bestuurbaar was met een externe controller. Hotswapping, audio en de controller zelf waren punten waar veel werk aan was, maar er was al iets bereikt.

Echter in deel 2, tussen evaluatie 1 en 2, was er geen echte vooruitgang geboekt… Er werd onderzoek gedaan in hoe ik de controller zou vervaardigen en dan beslist dat er gebruik zou worden gemaakt van de Arduino. Normaalgezien zou er via de GPIO pinnen een controller bevestigd worden die via een achtergrondprogramma steeds ondervraagd zou worden naar nieuwe informatie. Dit zou echter moeilijk op te stellen zijn. Lakka doet zeer moeilijk in het runnen van achtergrond programma’s, en alhoewel het mogelijk zou zijn heb ik weinig documentatie hierrond gevonden. Via een Arduino zou er een HID-verbinding tussen Arduino en Raspberry kunnen gebeuren. Deze zou zich kunnen voordoen als een game pad die bij druk op bepaalde knoppen een ander signaal doorsturen. Er werd dus een schema opgesteld en een 1e Eagle design opgesteld

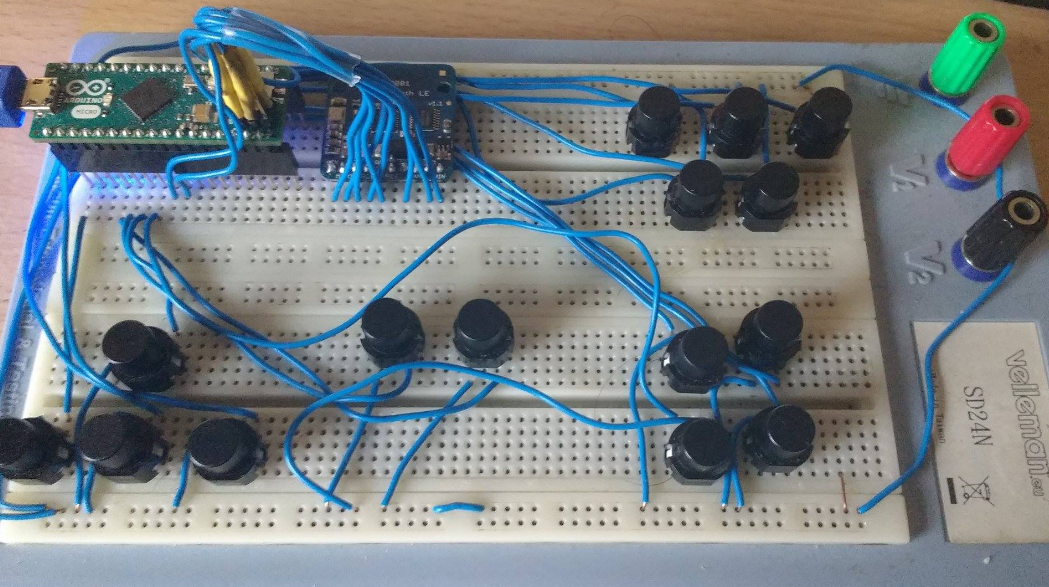
In het derde deel, het oorspronkelijke laatste, werd er dan meer gewerkt in de effectieve implementatie van de controller en werd hiervoor een werkend prototype gemaakt.

Het prototype bestaat uit een breadboard met een Arduino Micro en 15 drukknoppen die elk aan een pin en de ground verbonden zijn. Daardoor is er een eerder kleinere printplaat mogelijk. Deze wordt dan via de Arduino code gebruikt om de knoppen te vertalen naar game pad knoppen. Er was hier uiteraard wel wat werk rondom het debuggen.

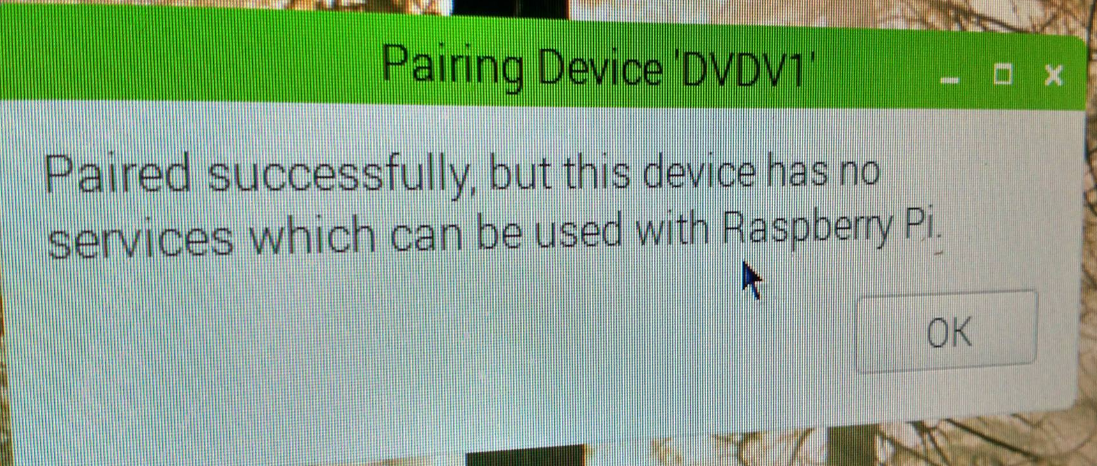
Daarna werd er gekeken naar het Bluetooth gedeelte. Deze was iets lastiger wegens de markt over gesatureerd was met diverse Bluetooth modules. Er werd gekozen voor de ADA-fruit RF8001 Bluetooth LE-module. Dit is een bluetooth module die leek te werken op Low Energy maar evengoed als normale Bluetooth, maar dit bleek toch niet het geval te zijn.

Bluetooth LE is een nieuw concept dat gebruik maakt van een andere connectievorm. Deze maakt echter enkel connectie bij het willen versturen vaneen signaal. Deze is dan ook niet eenvoudig te pairen met de meegeleverde Bluetooth antenne van de Raspberry Pi. Als deze echter ge-paired is zal deze in een slaaptoestand verkeren.

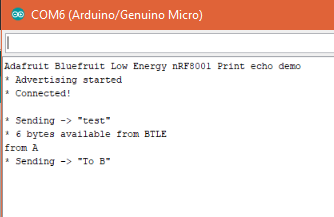
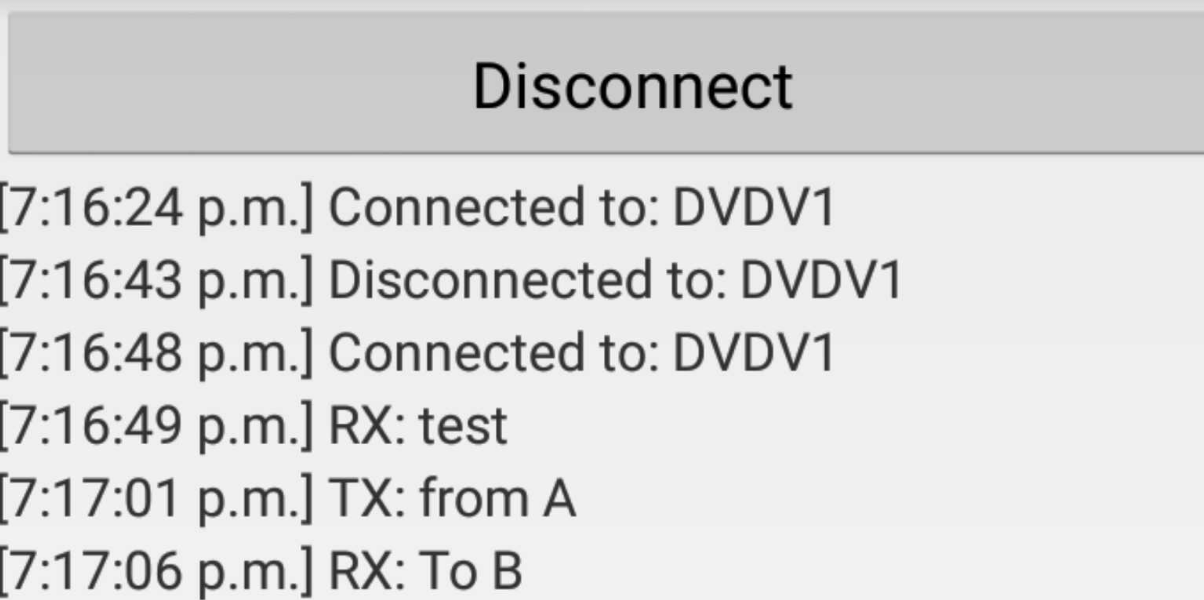
Het is wel al mogelijk om de drukknoppen uit te lezen op een ander toestel via de serial monitor.



In deel 4 is er verder onderzoek gedaan naar dit Bluetooth gedeelte. Deze is namelijk niet werkende geraakt tegen de eerste eindevaluatie. Er werd onderzoek verricht maar uiteindelijk bleek het simpelweg te liggen aan de Bluetooth vorm. Bluetooth LE is een vorm van Bluetooth die gericht is op energieconsumptie. De Raspberry Pi kan hier echter niet mee omweg en melde dit bij de uiteindelijke verbinding.



Met een tweede bluetooth module kan er connectie gemaakt worden tussen de 2. Deze werd dan getest en kon in combinatie met een 2e Arduino module de drukknoppen status doorgeven.

Dankzij de tweede Bluetooth module kan er een “draadloze connectie” gemaakt worden tussen de controller en de Raspberry Pi zoals gewenst. Echter denk ik persoonlijk dat er via een andere Bluetooth module (bvb de HC-05) wel een verbinding gemaakt zou kunnen worden, als deze dan ook uitgebreid werd met de mogelijkheid om zich als een toetsenbord te gedragen via zijn HID instelling, zou de 2e module en Arduino niet nodig zijn.

In deze fase werd ook de rapportering verbeterd en ingeleverd binnen de deadline.

# Zelfevaluatie

De zelfevaluatie in dit punt gebeurt op basis van het evaluatie formulier op Toledo.

* Nieuw verworven kennis: B

Bij aanvang van de opdracht wist ik bitterweinig over de Raspberry Pi. Het was puur gebaseerd op enkele foto’s op het internet waarbij anderen aantoonden dat het mogelijk was. Er is zelf onderzoek verricht om te weten hoe de Raspberry Pi werkt en hoe deze tot een werkende emulator kan omgevormd worden.

Er is in de loop van het project uitgebreid naar Arduino, waar er al reeds kennis van was, maar er ook een controller mee ontworpen werd. Het oorspronkelijke doel was om deze puur via hardware met de Raspberry Pi uit te lezen. Hierdoor is er kennis opgedaan van HID.

Echter ben ik geen expert geworden quasi Raspberry Pi. Ik ken het systeem nu voldoende om er andere projecten mee te ontwerpen maar daar stopt het dan ook. Ik zo bijvoorbeeld nog weinig van de GPIO pinnen, wat een zeer interessant deel lijkt van de Raspberry Pi.

* Toepassing aangetoond: B

De kennis die verworven is, is grotendeels te vinden in de praktische uitwerking van het project. Er zijn delen, zoals Bluetooth en Retropie waarbij ik persoonlijk denk dat er iets te weinig tijd is gespendeerd in de uitwerking hiervan, maar beide zijn uiteindelijk wel werkend. Bluetooth is zo aangepast dat er met 2 modules gecommuniceerd kan worden i.p.v. met een enkele module en de Raspberry pi’s ingebouwde antenne.

Er is uiteindelijk binnen het project gebleven en er zijn geen uitbreidingen toegepast.

* Planning: C

In het project is er een begin-planning opgesteld maar deze werd niet behaald. Er werd eerder gewerkt wanneer ik tijd en zin had, wat leidde tot een snelle en vlotte start, maar een zeer traag einde. Persoonlijk ben ik nooit een persoon geweest die zich makkelijk aan een planning kan houden, maar de iets te hoog gekozen doelstellingen zorgden ervoor dat als vooruitgang traag verliep de moed snel zakte.

Planning en rapportering is zeker een verbeteringspunt.

* Analyse noden: C

Er was duidelijk een tekort bij de planning van de te verwerven kennis. Het feit dat er een extra microcontroller gevraagd was voor een controller en er veel meer kwam kijken bij de Bluetooth verbinding hebben dit duidelijk aangetoond.

* Initiatief: B

Persoonlijk vind ik dat er voldoende werk is gegaan in het project. Er is vrij veel bereikt voor hoe diepgaand het onderwerp bleek te zijn. Er werd enkel bij aanvang een verkeerde schatting gemaakt van de diepgang en uitdaging van het project.

* Rapportering: C

De rapportering was zoals bij de eerste examenkans vermeld te weinig. De eindresultaten waren amper beschreven, en het document zelf was niet gecontroleerd op spellingsfouten.

De tussentijdse rapporteringen waren ook vaak een bijkomstig punt terwijl deze eerder als een leidraad diende te zijn.

# Inhoud van het portfolio

De inhoud van het portfolio bestaat uit:

* Een installatie handleiding
* Een simpele TXT handleiding van hoe je het best een SD kaart backupt en herformateerd.
* De buttonlayout van het project, dit met pinning.
* De gebruikte datasheet van de Arduino Micro
* Arduino code die gebruikt werd om bluetooth verbindingen mee te maken, de drukknoppen te testen, een toetsenbord HID test, de werkende A (controllerside) code en de werkende B (Raspberryside) code voor de bluetooth met 2 Arduino’s.
* Tevens zijn er eagle schema’s gemaakt rond de 2e fase, maar deze zijn ondertussen ietwat verouderd. Er zou eerst gekeken worden naar de bluetooth uitbreiding.

# Bronnen

|  |  |
| --- | --- |
| [1] | [Online]. Available: https://www.raspberrypi.org/help/videos/. |
| [2] | [Online]. Available: https://nl.wikipedia.org/wiki/Raspberry\_Pi. |
| [3] | [Online]. Available: http://superuser.com/questions/17350/whats-the-life-expectancy-of-an-sd-card. |
| [4] | [Online]. Available: https://nl.wikipedia.org/wiki/Live-system. |
| [5] | ebay, „Ebay; sega mastersystem prijs,” Ebay, [Online]. Available: http://www.ebay.com/itm/SEGA-Master-System-Console-MK-2000-Two-Pads-PSU-RF-switch-Manual-Boxed-set-B/152665785672?hash=item238b970548%3Ag%3AnLgAAOSw-u1ZeaoM. [Geopend 20 8 2017]. |
| [6] | [Online]. Available: https://www.raspberrypi.org/learning/hardware-guide/components/raspberry-pi/. |
| [7] | [Online]. Available: https://github.com/raspberrypi/noobs/blob/master/README.md. |