

Realisatieplan

PLONG

Auteurs: Donovan Khounm, Floris Mesu, Mark van Oirschot,
Merlijn van der Veen & Piotr Tadrata

Inhoud

| | |
|---|----|
| Inleiding | 2 |
| Narrative & concept | 3 |
| Brand Style Guide | 4 |
| User flow | 5 |
| Onderdelen van PLONG | 7 |
| Controller | 12 |
| Hoe is de controller bepaald..... | 12 |
| De onderdelen van de kattenstaart | 13 |
| Display matrix | 15 |
| Kostenplan | 18 |
| Hoe zet je het prototype in elkaar? | 19 |
| Hardware diagram | 21 |
| Advies voor vervolg-onderzoek..... | 22 |

Inleiding

Om het prototype tot stand te brengen, hebben we ons het afgelopen semester beziggehouden met verschillende losse onderdelen. In dit document worden zowel de gemaakte keuzes voor de controller, narrative, het display en de microcontroller toegelicht, als de functionele hardware diagrammen, zodat het mogelijk is om het prototype te reconstrueren. Ook wordt er advies gegeven over vraagstukken die eventueel in het volgende semester onderzocht kunnen worden om het prototype verder uit te werken.

Narrative & concept

The narrative

Making use of Plant-E technology, users play PONG, being displayed at GLOW, goals is to make people feel connected with nature. The goal of connecting with nature, is to serve being Zen. As someone who attends the festival, you get bombarded with so many things. Lots of information. All those lights. All those sounds, all those people. You are practically overwhelming your nervous system. By making people play PLONG, a plant (low)powered, simple PONG game, with dreamy visuals amidst the bustling crowd, we aim to create that Zen Garden for the people. You just must interact with 1 other person and are invited to look deep within yourself and notice your heartbeat. Calm down and play with nature, let yourself be taken down the stream and hear the sound " PLONG ".

The concept

An art display powered by Plant-E technology and it being low energy. Before interacting with the hardware setup, the users partake in a guided meditation. This will allow them to play the game in a better mindset. The game displayed on the matrix is the game PONG. The users interact with it using a lever. The goal is to feel zen.

Tone of communication

Calm and collected. We do not want long and wordy phrases. Just keep it short and simple. Effective and leaving lots of space for silence. We aim to communicate the feeling of Zen to the user. Omitting unnecessary words and getting straight to the point. Haiku forms would be nice to play into the Japanese culture. The cadence of those poems also helps along with the restriction of number of words.

Brand Style Guide

Brand Style Guide

Typography

Specimen

BRICKTOWN







ABCDEFGHIJKLM

NOPQRSTUVWXYZ

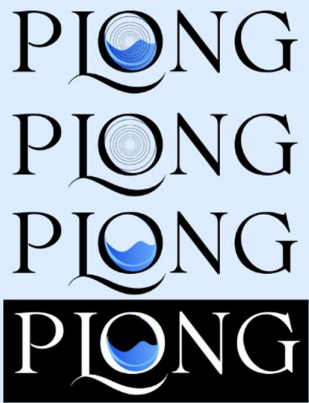
ABCDEFGHIJKLM

NOPQRSTUVWXYZ

Color Palette

| | | |
|---|---|---|
|  |  |  |
| #000000 | #4d8bf5 | #255eed |
|  |  |  |
| #ffffff | #748b75 | #503d42 |

Logo variations



User flow

De user flow voor PLONG is niet geheel complex. Zijnde dat er een er vier main loops zijn die zich afwisselen per game status. Deze game loops zijn de twee start-up loop – verdeelt over de twee spelers-, de main game loop en de maximale score reset loop.

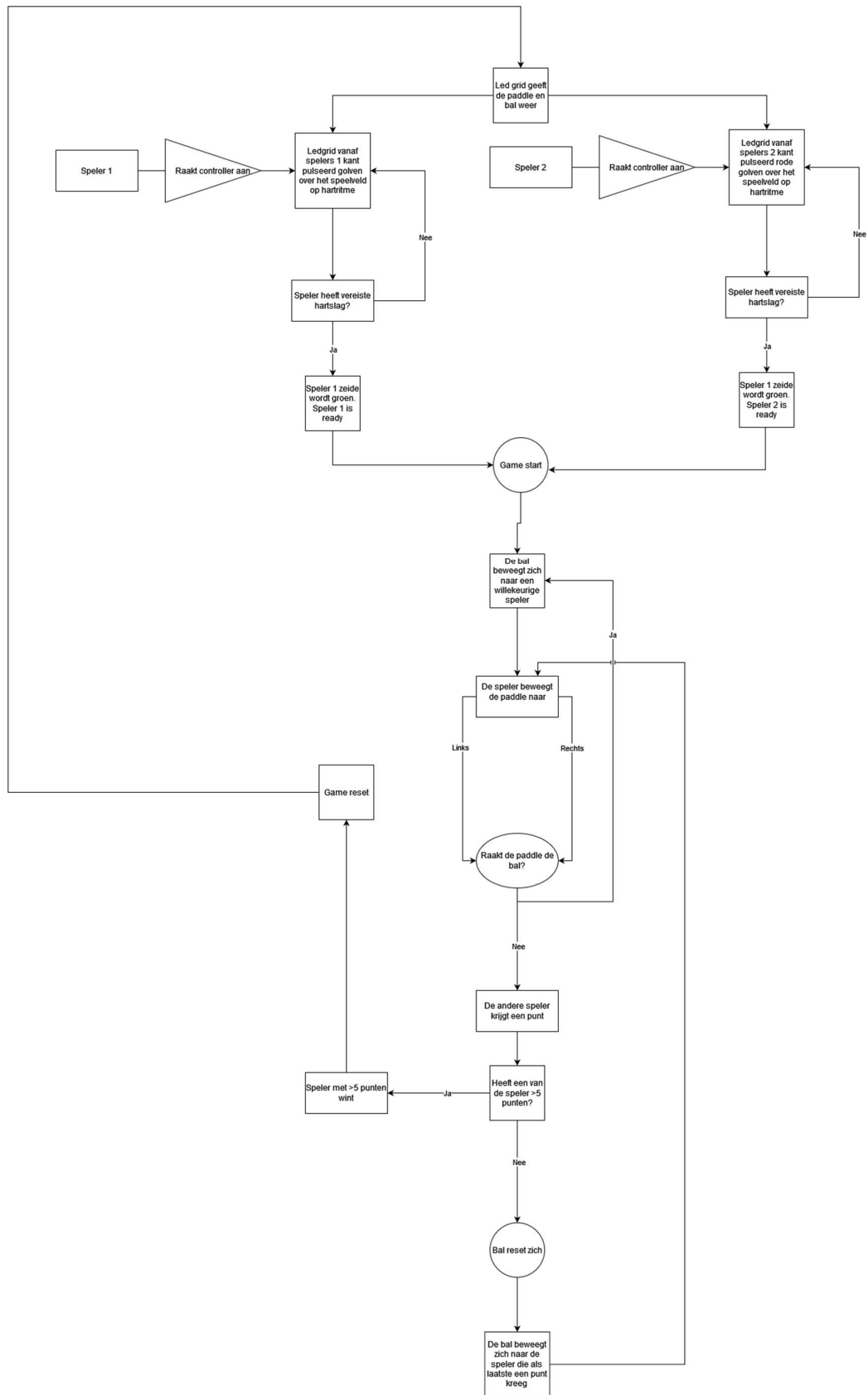
De start-up loop bepaald of de game wel of niet start. De game checkt, zodra een of meerdere spelers de controllers aanraken, of de hartslag van de speler laag genoeg is om de game te starten. Is dat niet het geval? Dan start de game niet. Is dat wel zo? Dan kan de game beginnen als dat voor beide spelers geldt.

Of de juiste hartslag is bereikt, wordt gecommuniceerd doormiddel van een rood of groene kleuring op het ledgrid. De kleuring wordt op de paddle en op geanimeerde golven toegepast. Als de speler een te hoge hartslag heeft, dan kleurt het rood aan de kant van die speler. Is deze echter laag genoeg, dan kleurt het groen. De speler is dan klaar om te spelen.

Zodra de game is gestart, begint de main gameloop. Deze start zodra de game de bal naar een willekeurige speler stuurt. De speler kan dan de paddle bewegen om de bal te onderscheppen. Lukt dat? Dan kaatst deze naar de andere speler en blijft deze loop draaien. Het is dan aan de andere speler om de bal weer terug te kaatsen.

Lukt dat niet? Dan krijgt de andere speler een punt. De bal reset zich en beweegt dan naar de speler die als laatste een punt heeft gekregen. Deze loop blijft draaien totdat een van de spelers de maximale aantal punten van 5 heeft.

Zodra de het maximale punten aantal is bereikt, dan start de maximale score reset loop. De game geeft aan welke speler heeft gewonnen door aan die kant te knippen. Daarna reset de game zich en begint de game weer vanaf voor aan. Op de volgende pagina staat de gameloop schematisch weergegeven.



Onderdelen van PLONG

De onderdelen op het bord zijn een

- Heartbeat sensor (MAX30100).
- Schuifpotentio meter van 10kΩ.
- ESP32 is de microcontroller.
- TCA9548a
- Bevestigingsmateriaal.

Micorcontroller:

De eerst gekozen micro controller was de Arduino Nano dit is gekozen om zijn lage energie verbruik en goed genoeg voor een klein project. Tijdens prototyping kwamen we erachter dat deze microcontroller niet genoeg werkgeheugen heeft voor ons pong project. De Arduino NANO heeft een werkgeheugen van 2kb.

```
Warning! The data size (2083 bytes) is greater than maximum allowed (2048 bytes)
RAM: [=====] 101.7% (used 2083 bytes from 2048 bytes)
Flash: [==    ] 22.3% (used 6864 bytes from 30720 bytes)
Building .pio\build\nanoatmega328\firmware.hex
```

De code die we hebben gebruikt te veel werk geheugen iets wat ervoor zorgt dat de controller crasht. We hebben daarom ervoor gekozen om als alternatief de ESP32 WROOM te gaan gebruiken. Dit omdat dit het meest voor de hand ligt en deze dezelfde specs bevatten als de arduino nano alleen heeft de ESP 32 een werkgeheugen van 32 kb.

```
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=       ] 7.4% (used 24252 bytes from 327680 bytes)
Flash: [==    ] 21.7% (used 284353 bytes from 1310720 bytes)
Building .pio\build\my_esp32_project\firmware.bin
```

Het nadeel hieraan is dat de esp32 een hoger energie verbruik heeft. Dat heb ik gemeten bij het onderzoeksrapport "Onderzoeksrapport Glow". Dat is zo ongeveer 10 keer zo hoog. De stakeholder heeft aangegeven dat het voor nu zichzelf niet zoveel wil bezighouden met het energie verbruik maar meer met de Ux. Wanneer de code is geschreven kan men nog eens gaan onderzoeken wat nu een geschikte controller is die voldoet aan de energie reductie alleen voor nu ligt dit buiten de scope en is dit de gepaste oplossing.

| | |
|----------------------|--|
| Stm32F407 | $5 \text{ volt} - 48\text{mA} = 5 * 0,048 = 240\text{mW}$ |
| Z80 + AT28C256 | $5 \text{ volt} - 11\text{mA} = 5 * (0,001 + 0,010) = 55\text{mW}$ |
| CMOS 6052 + AT28C256 | $5 \text{ volt} - 10\text{mA} = 5 * (0 + 0,010) = 50 \text{ mW}$ |
| Arduino NANO 328 | $5 \text{ volt} - 13\text{mA} = 5 * 0,013 = 65\text{mW}$ |
| ARDUINO MEGA 2560 | $5 \text{ volt} - 35\text{mA} = 5 * 0,035 = 175\text{mW}$ |
| Esp8266 | $5 \text{ volt} - 79\text{mA} = 5 * 0,079 = 395\text{mW}$ |
| Esp32 | $5 \text{ volt} - 122\text{mA} = 5 * 0,122 = 610\text{mW}$ |
| Raspberry pi 4 | $5 \text{ volt} - 406\text{mA} = 5 * 0,406 = 2003\text{mW}$ |

Potmeter:

Er is gekeken naar een gepaste oplossing voor het bewegen van de joystick de meest voor de hand liggende en enige oplossing is de potmeter. Dit een regelbare weerstand en in combinatie met een analoge ingang kun je er data mee inregelen.

De potmeter die gekozen is een schuifpotmeter (mogelijk 2 kanalen) van $10k\Omega$. Dit is dezelfde soort potmeter die bij het prototype is gebruikt alleen dan door te draaien ([draai potmeter](#)). Met deze schuif potmeter kan men op de analoge ingang van de Esp32 de ingang signalen ontvangen en vertalen naar data. De esp32 heeft 4095 bits analoge ingang dat weer veel nauwkeuriger is dan de 1023 bits analoge ingang van de Arduino Nano.

De System test video heb ik in bijlage erbij gezet. Dit is een met de esp32 en Arduino Nano waarin ik met een simpele code het effect van de potmeter laat zien.

Tijdens het gebruiken van ons pong spel zagen we dat als je de potmeter van speler 1 heel hard draait dat je de paddle van speler 2 ook laat bewegen. We hebben hier even lopen sparren met een docent die het ook heel raar gedrag vond (Lennart). Er waren 2 mogelijke oorzaken 1) De voeding van de ledverlichting zit ook aangesloten op de potmeter. 2) Het zou ook kunnen dat wanneer de spanning te hard wisselt door de potmeter te hard te draaien dat het gaat storen op je circuit. Oplossing 1 was om de potmeter zijn voeding dichtbij de voeding van de lab voeding te zetten. Oplossing 2 was om tussen de potmeter een condensator te zetten van 2 Nano farad. Er is gekozen voor oplossing 2 omdat deze het makkelijkst was te realiseren. Dit had uiteindelijk ook een positief effect en de errors zijn niet meer voorgekomen.



Heartbeat Sensor.

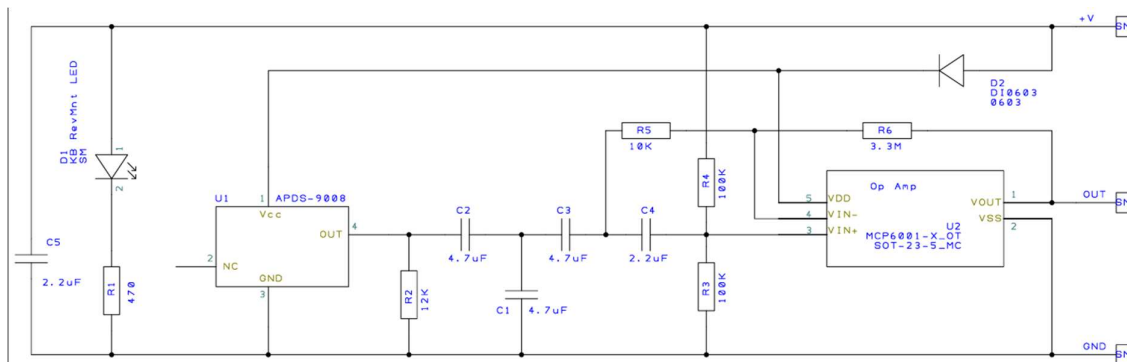
Het eerste waar de groep mee bezig is geweest is met de [MAX30100](#) een heartbeat sensor die werkt op een Communicatieprotocol van I2C. Dit vonden wij een goede oplossing alleen merkte we op dat dit component een fixed adress heeft.

The MAX30100 slave ID consists of seven fixed bits, B7-B1 (set to 0b1010111). The most significant slave ID

We hadden 3 oplossingen.

| | |
|----|--|
| 1. | Een I2C multiplexer om zo telkens te switchen van het lezen van je sensor. |
| 2. | Een microcontroller te gebruiken met twee I2C lijnen. Een Arduino Mega2560 heeft er namelijk twee opzitten die je dan eventueel beide apart aan zou kunnen sturen. |
| 3. | Een nieuwe heartbeat-sensor gaan zoeken. |

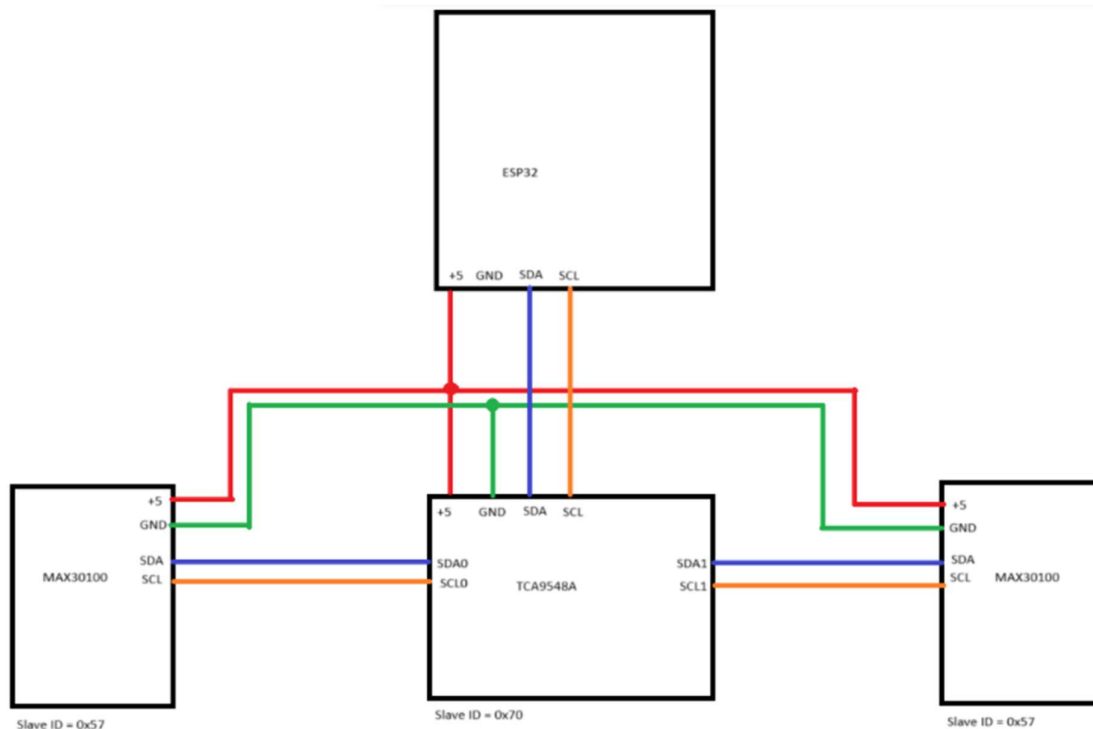
We hebben gekozen voor oplossing 3 het zoeken van een andere heartbeat sensor. Dit is de [pulse sensor](#). Het heeft geen moeilijke technologie alleen een Led die op je huid schijnt en een ontvanger die kijkt naar de pulses in je vingers (APDS-9008) en een op-amp die het compatible maakt voor de analoge ingang. Dit klonk als een ideale oplossing omdat we hiermee het hadden opgelost we hadden nog meerdere analoge ingangen op onze microcontroller vrij.



Het probleem is dat de calculaties maken met dit component moet het gebruik maken van een interrupt timer die in de microcontroller zit. Bij de esp32 zijn hier twee van aanwezig het is best een uitdaging om deze software aan te passen in je source code. Daarna is nog gekeken of er een eigen library kon worden gemaakt op basis van de gegevens die ik binnenkrijg op de analoge input alleen wisten we daar niet echt raad mee.

| | | | |
|----|----------|---------|---|
| 15 | Internal | Timer.1 | 3 |
| 16 | Internal | Timer.2 | 5 |

Daarvoor is geraadpleegd bij een expert (Edwin) om met mij mee te kijken naar een oplossing. Edwin vertelde dat het niet verstandig was om deze pulse sensor te gebruiken omdat dit nogal een onnauwkeurige pulse sensor is en dat hij zelf geen voorstander was van analoge meters. We hebben over ideeën gediscussieerd en gekozen om de MAX30100 te gaan gebruiken met een multiplexer ([TCA9548A](#)). Dit zal uiteindelijk ook voor de beste resultaten bieden omdat dit kwalitatief een betere chip is.



Deze installatie was eerst op een breadboard gemaakt zodat er makkelijk kon worden gekeken wat de juiste code hiervoor was. Hiervoor heb ik een langere tijd op lopen debuggen hoe de heartbeat sensoren werkend konden worden gemaakt in combinatie met de esp32 en de multiplexer. Methodes voor het oplossen waren.

- Eerst de heartbeat sensor alleen aansluiten.
- Een heartbeat sensor aansluiten met een kanaal van de multiplexer.
- De gehele installatie maken met een andere microcontroller (Arduino Mega)
- Oscilloscoop gebruikt om de signalen te meten.

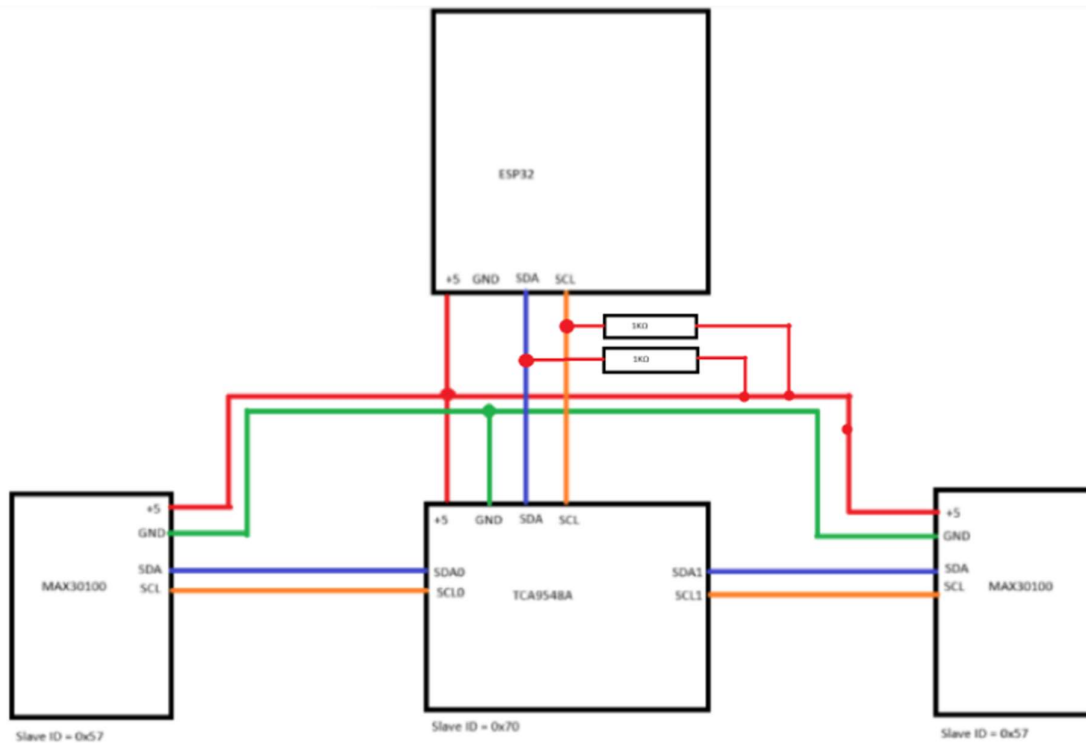
Toen de installatie op een arduino Mega 2560 aan was gesloten. Kon het in ieder geval de leds aanzetten van de sensoren. Toen ben ik de signalen gaan uitlezen met een oscilloscoop het was op zich niet verrassend dat de controller een ander level van communicatie gebruikte.

- Arduino Mega communicatie level is +5v
- Esp32 communicatie level is 3,3v.

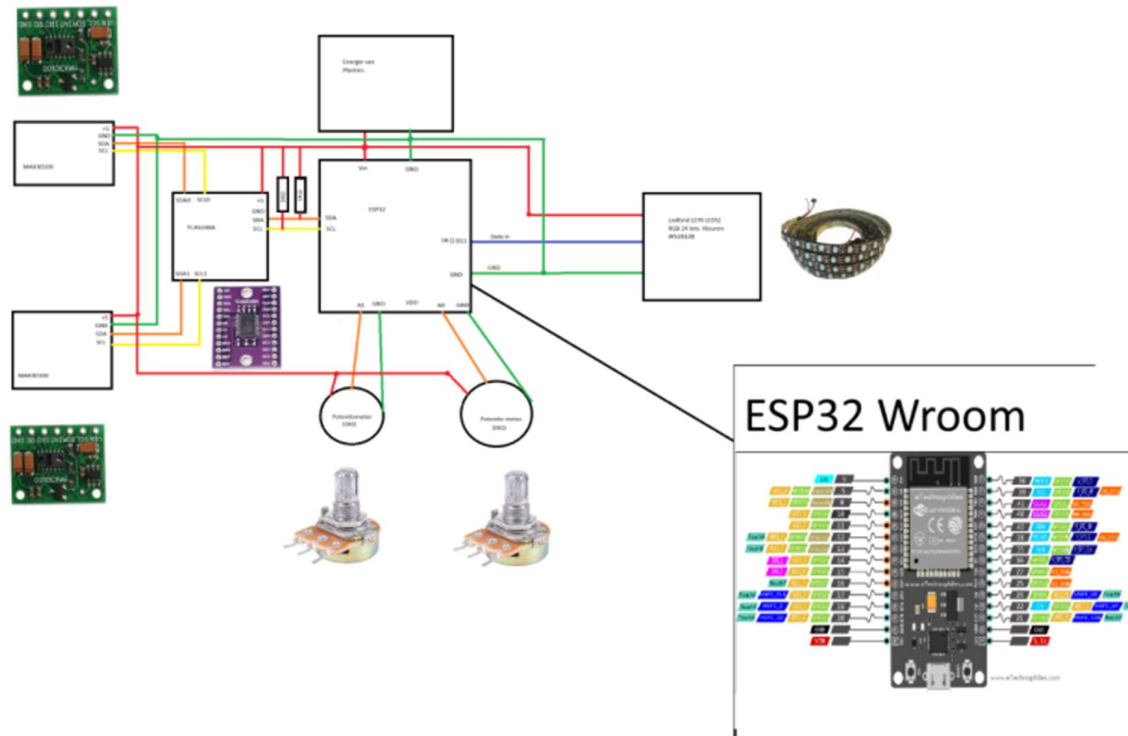
Volgens de datasheet zou dit niet moeten uit maken en kan de TCA9548A hier prima mee om kunnen gaan.

- Allows voltage-level translation between 1.8-V, 2.5-V, 3.3-V, and 5-V buses

Alleen in de praktijk bleek dit niet zo te zijn. Uit experiment is er toen gekozen voor het gebruik maken van Pull up Weerstanden van 1 K Ω een andere optie zou nog kunnen zijn is een [TXS0108E](#) Logic Level converter om de communicatie van 3,3 level naar een hoger level te zetten +5v. De pull up weerstanden zijn gebruikt deze zorgen ervoor dat de communicatie nu hoog genoeg is om de multiplexer en de heartbeat sensoren aan te kunnen sturen. Dit wordt als de eindoplossing gebruikt omdat dit de makkelijkste oplossing was en voldoende is.



Eindproduct



Het bevestigings materiaal dat is gebruikt is een breadboard om de esp32 en de multiplexer in te plaatsen. De draden die er op zijn gesoldeerd zijn soepel zwarte draad van 1mm dik.

Controller

De controller moest aan verschillende requirements voldoen. Dit waren:

- Past de controller bij de gebruiker
- Past de controller bij de locatie
- Is de controller energiezuinig?

Via deze requirements is er bepaald hoe de controller eruit ging zien.

Hoe is de controller bepaald

De controller is bepaald doormiddel van een onderzoek. Daarin is er gekeken naar de gebruiker en welke digitale inputs zij fijn vinden. Dit is gedaan door te bepalen welke groep er naar GLOW komt, waarna er een enquête is verspreid onder deze mensen.

Naast de gebruikers is er ook gekeken welke inputs passen bij GLOW en haar omstandigheden. Zijn deze stevig? Kunnen ze waterdicht gemaakt worden? Hierdoor kunnen we ervan uitgaan dat de gekozen inputs GLOW zullen overleven.

Daaruit zijn verschillende inputs naar voren gekomen die passen bij ons project. Met deze inputs zijn dan verder controller concepten gemaakt, waarvan twee tastbare mock-ups zijn gemaakt. Dit waren de kattenstaart en de draaibare stronk.



Als groep, samen met de student expert die op GLOW heeft gestaan, hebben wij de keuze gemaakt om de kattenstaart verder uit te werken. Dit hebben wij zo besloten omdat:

- De controller is opvallend.
- Hij is simpel qua opzet
- Hij past goed binnen het beoogde thema
- Er zijn weinig breekbare onderdelen

De draaibare stronk viel af omdat deze wat minder goed binnen het thema paste. Daarnaast was deze ook moeilijk te integreren in de opstelling die wij voor GLOW voor ogen hadden. Dat kwam omdat dit een controller is die op een verhoogd stukje moet komen te staan. Daar had de kattenstaart minder last van.

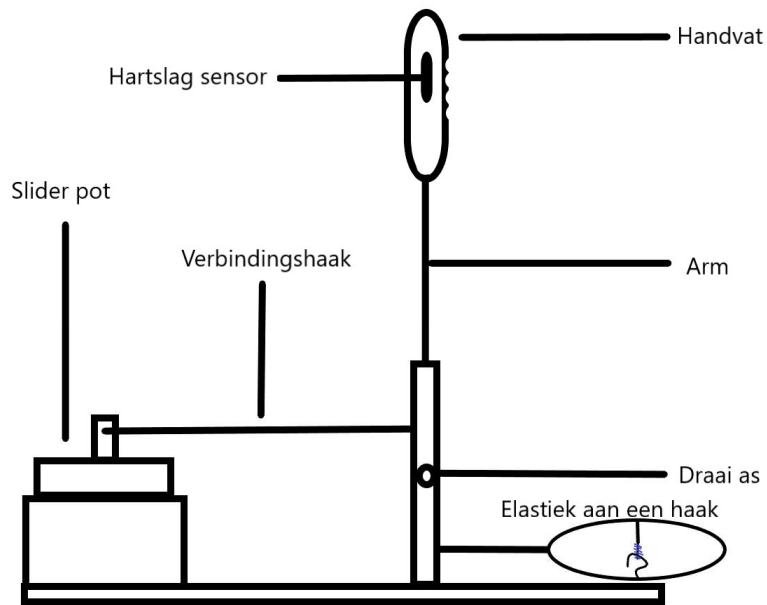
De onderdelen van de kattenstaart

De kattenstaart bestaat uit een aantal onderdelen. De technische onderdelen zijn in het vorige paragraaf besproken. Hier zal echter wel benoemd worden welke van die onderdelen in de controller terecht komen.

De kattenstaart is opgebouwd uit drie groepen:

- De grip
- De arm met een draai-as onderaan. Aan de grond verbonden met elastiek
- Een slider potentiometer die met een haak is verbonden aan de arm.

De hele groep ziet er schematisch uit als onderstaand:



Hieronder wordt elk onderdeel groep beschreven. Hier komen ook per onderdeel benodigde specificaties als lengte aan bod.

De grip/handvat

De grip is in de vorm van een kattenstaart. Aan de linker kan zitten vier dallen waar de vingers in passen. Een 45 graden naar recht daarvan zit een verlaagd stuk met een balk waar de hartslagmeter in past. Deze kan doormiddel van een gat dat doorloopt naar de buisholte de kabels van de hartslagmeter via de arm naar beneden voeren.

De grip heeft een diameter van 4 cm. Hij is 16cm hoog. In het midden, aan de onderkant van de lengte van de grip, zit een gat met een diameter van 1,7cm. De grip is gemaakt met een pvc-buis van 1,6 cm in het achterhoofd. De grip 0,1 cm speling waardoor relatief los erop zit. Bij verdere development raad ik aan om, zover dat mogelijk is, de diameter kleiner te maken. Echter was ik gelimiteerd door de nauwkeurigheid van de 3d printer en niet meteen kunnen testen of deze over de arm zou passen.



De arm en draai-as

De arm dient als het beweegbaar element dat de beweging van de gebruiker doorgeeft aan de rest van de controller. De arm die wij gebruikt hebben is een pvc-buis van 1,7 cm. De lengte is ongeveer anderhalve meter. De arm zit vast aan de controller opstelling doormiddel van een draai-as onderdaan de lengte van de arm. Hierdoor kan de arm naar links of rechts gedraaid worden.

Onderaan de arm, in de basisplaat waar de controller op gebouwd is, zit een haak en elastiek. De elastiek dient als kracht die de arm terug naar het nulpunt trekt. Als de controller naar links geduwd wordt, zal de elastiek ervoor zorgen dat deze automatisch weer in het midden komt. Daarnaast zorgt de elastiek er ook voor dat de controller rechtop blijft staan als deze niet gebruikt wordt.

De slider potentiometer

De slider potentiometer is het component dat de beweging van de draaiarm registreert en terugstuurt naar de computer die Plong aanstuurt. De potslider is verbonden met een haak aan de draaiarm.

Wanneer de draaiarm naar links getrokken wordt, duw de haak de verbonden potslider naar links. Gaat deze naar rechts, dan wordt deze weer naar rechts getrokken. In rust staat de potmeter altijd in het midden. Dit omdat de draaiarm naar het midden getrokken wordt door de elastiek.

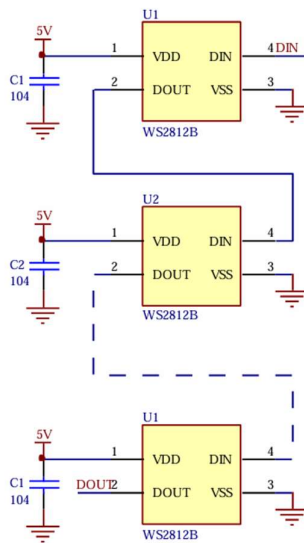
Display matrix

Vanwege Ux wilde men graag een led grid met verschillende kleuren maar ook kleuren die je harder kon laten branden en zachter kon laten branden. We hebben uiteindelijk gekozen voor een Ledstrip van de [WS2812b](#).

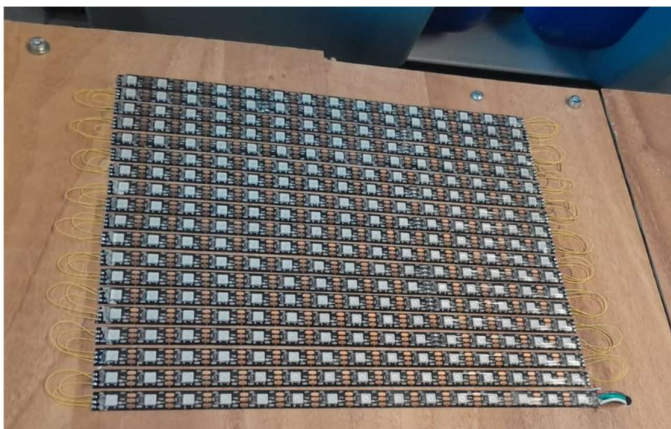
Deze ledstrip bevat 24 bits aan kleuren dat betekent 16.777.215 kleuren daarmee konden we het project gaan maken. Het heeft een interne chip per led waardoor we maar 2 uitgangen van de microcontroller hoeven te gebruiken(Datapin, GND). Het enige nadeel hieraan is dat ze hoog in verbruik zijn alleen hoeven niet alle leds aan te zijn tijdens het spelen van pong wel

wordt aangeraden om een externe voeding van +5 volt te gebruiken omdat de microcontroller zelf dit niet aan kan.

Er is een led strip besteld van 300 leds die we hebben opgesplitst in 19 lijnen van 15 leds breed. Deze hebben we tot een led matrix weer aan elkaar gesoldeerd. Aan het begin van het maken verwachtte we meerdere datalijnen nodig te hebben vanwege snelheid waarin een Expert (Lennard) ons had verteld dat een datalijn snel genoeg is wat achteraf ook zo bleek te zijn. Hoe de leds zijn aangesloten is goed te zien op de applicatie note van de leverancier.



Iedere led heeft een eigen mini controller die alle kleuren handeld. De led krijgt een vaste voeding van +5 volt versterkt door een condensator en een vaste Ground. De led zelf staat altijd aan. Door alle leds loopt een Data in (DIN) en een Data out (DOUT). Daarmee kun je de controllers besturen over deze lijn stuur je je signalen overheen.



Deze ledstrip zou ook gebruikt kunnen worden in het eind project. Waarbij men meerdere strips bij elkaar maakt waardoor het dient als een lamp. Ik heb een filmpje geplaatst waarbij ik aantoon dat wanneer je de leds 3x3 doet er veel licht vanaf komt en dat we dit dan ook

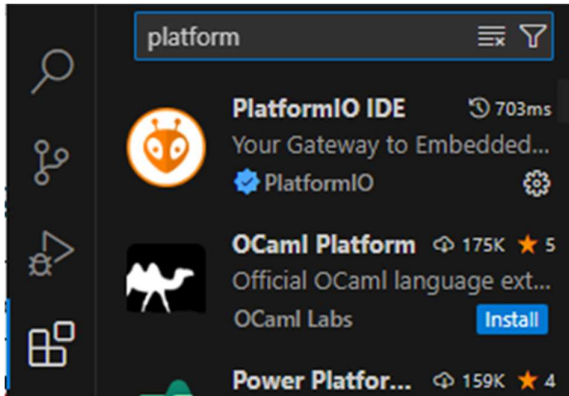
kunnen toepassen wanneer het project wordt gerealiseerd. Zodat men de verlichting van een verre afstand ziet bewegen.

Kostenplan

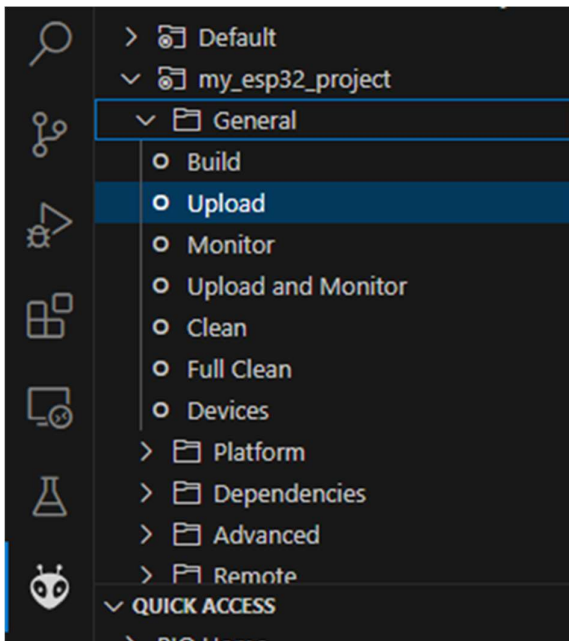
| | |
|---------------|----------------|
| Led strip | € 36,85 |
| ESP 32 | € 12,95 |
| MAX30100 | 2x € 7,92 |
| Potmeter | € 8,60 |
| TCA9548A | € 3,50 |
| Totaal | € 77.74 |

Hoe zet je het prototype in elkaar?

Om het prototype in elkaar te kunnen zetten, heb je eerst de firmware nodig die op de controller zal draaien. Deze kun je [hier](#) downloaden. Zodra je de repository lokaal hebt opgeslagen, kun je deze openen in Visual Studio Code. Om de ESP32 met de code te kunnen flashen, heb je de PlatformIO IDE-plugin in Visual Studio Code nodig. Deze kun je via de plugins-tab downloaden.

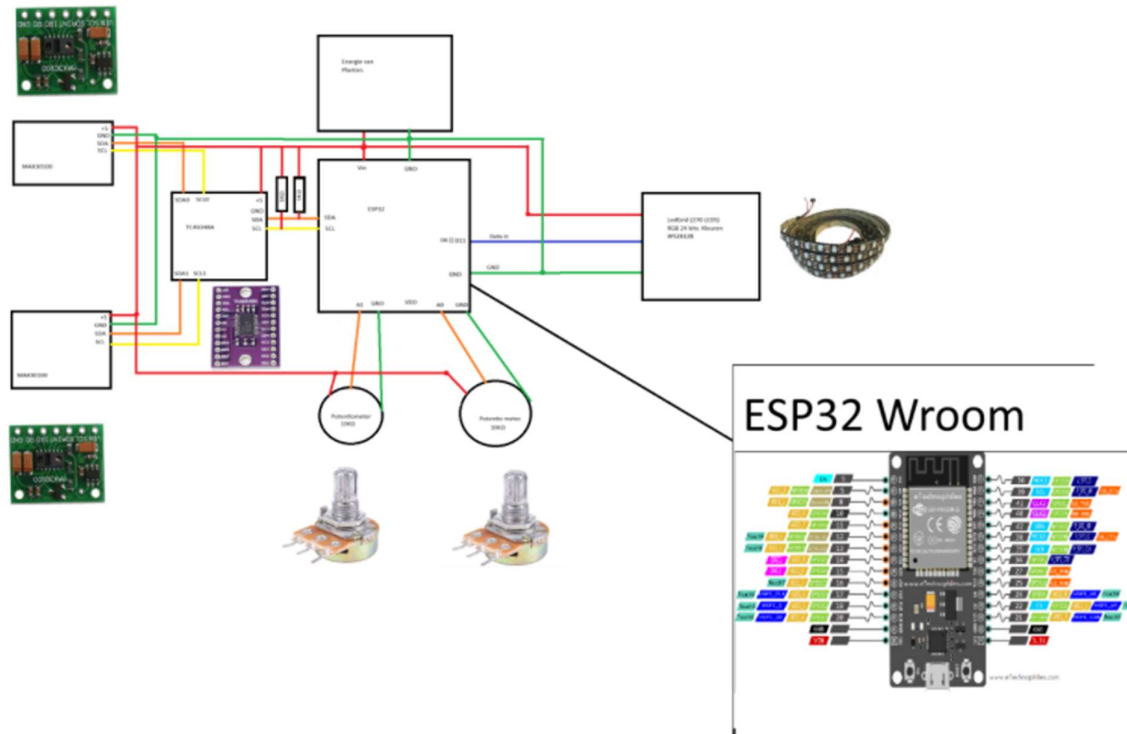


Zodra PlatformIO geïnstalleerd is, kun je via de PlatformIO-tab de repository selecteren en uploaden naar de ESP32.



Zodra Visual Studio Code klaar is met flashen, is de microcontroller klaar om aan de rest van de hardware te worden aangesloten. Kijk hiervoor naar het hardware diagram.

Hardware diagram



Ons hardware diagram is een installatie die wordt gevoed door een (labvoeding). De labvoeding simuleert hierbij de stroom die van de planten af zou moeten komen. De controller die is gebruikt is de "ESP32 Wroom" die de ledmatrix aanstuurt. De potmeters handeld de paddles. De heartbeat sensoren zorgen dat het spel wordt gestart. Tussen de heartbeat sensoren zit een I2C multiplexer deze zorgt ervoor dat de communicatie iedere keer wisselt van kanaal.

Advies voor vervolg-onderzoek.

- Hoe krijgen we een installatie op Glow dat ook (spat)waterdicht is.
- Is er een energiezuinigere microcontroller op de markt.
- Wat is de invloed op het spel als de installatie langere draden nodig heeft.
- Is I2C wel een gewenst protocol voor een grote Controller. Is in dat geval CAN of RS485 ook een mogelijkheid vanwege mogelijke storingen door het bewegen.
- Hoe gaan we de groene spanning reguleren naar een vaste voeding van +5 volt.
- Zijn er nog meer features die we kunnen toevoegen voor het gebruik van de hartslag sensor.
- Waar liggen kansen voor meer energiezuinigheid in het project.
- Is Plant-E technologie de enige manier voor het opwekken van groene energie?