



3D printer

Author *Gijs Jackers*
Bo Mengels
Martijn Guilliams

Intern *Koen Gilissen*
Ward Martens

Abstract

Het automatiseren van een print queue voor een 3d-printer maakt het voor studenten van de professionele bachelor Elektronica-ICT een moeiteloze zaak om prototypes te printen.

Studenten moeten vaker en vaker een klein onderdeel printen om eens iets uit te proberen of voor een finaal product. Het probleem valt dan echter dat het printen vaak lang duurt en het wachten om dan zelf een print op te zetten is een onbegonnen zaak. Een oplossing daarvoor zou kunnen zijn is meer 3d-printers kopen. Dit is een dure en zelfs dan nog niet een zekere oplossing.

Heel veel tijd gaat er eigenlijk aan verloren door de tijd tussen twee prints omdat er bijvoorbeeld geen melding is en de volgende persoon onmogelijk altijd bij de printer klaar zit om zijn/haar print op te zetten. Ook gaat er tijd verloren omdat er 's nachts na de schooluren niks meer kan worden opgezet. Een gedeeltelijke oplossing hiervoor zou zijn om een print van acht uur of langer in te plannen om op het laatste moment van de schooluren op te zetten. Maar dit is ook geen volledige oplossing voor dit probleem.

De complete oplossing voor dit is door gebruik te maken van een Raspberry Pi met een Octoprint installatie. Deze installatie zal dan een queue runnen. Deze software kan een print dus ook automatisch opzetten en een script runnen om het bed te clearen. Zo is er geen enkel moment dat de printer onnodig stilstaat omdat deze niet beschikbaar zou zijn.

Dit zal ervoor zorgen dat studenten geen onnodig tijd verspillen.

Content

1	Introduction.....	3
2	Material and methods	3
3	Results	4
3.1	Raspberry Pi verbinden met 3D printer.....	4
3.2	PrusaSlicer	4
3.3	Raspberry Pi tijdzone	4
3.4	Firmware updaten van 3d-printer	5
3.5	Bumper	5
3.6	Camera en camera staander	7
3.7	Toegangsgroepen maken	8
3.8	Plugins.....	8
3.8.1	Display layer progress.....	8
3.8.2	Print time genius.....	8
3.8.3	Filament manager.....	8
3.8.4	Octo text	9
3.9	Continu printen	9
3.10	Raspberry Pi via ssd	9
3.11	Bed clearing	10
3.12	Log bestanden	10
3.13	Wan verbinding	11
3.14	Script gebruikers.....	11
4	Discussion	11
5	Conclusion	12
6	Reference list.....	13

1 Introduction

De opdracht voor Project Embedded was om een 3d-printer volledig automatisch te laten printen met een ingebouwde wachtlijst. Dit moest veilig en overzichtelijk gebeuren. Met overzichtelijk wordt er bedoeld dat er ook bijgehouden kon worden wie wat heeft geprint. Er moest dus een service aangeboden worden voor studenten, zodat ze gemakkelijk iets konden printen zonder te moeten wachten op een beschikbare printer. Ze moesten dus ook van thuis uit kunnen printen. De website is enkel beschikbaar voor studenten van Elektronica-ICT.

2 Material and methods

De printer die gebruikt wordt voor dit project is de Prusa i3 MK3S+. De 3D printer is verbonden met een Raspberry Pi 4. Op deze pi zal de server (octopi) draaien en op de SSD die gekoppeld is aan de Raspberry pi zullen de bestanden van de queue geschreven worden. Octopi is de naam van de server die voor dit project gebruikt wordt en deze is gebaseerd op de open source versie OctoPrint. Hieronder is nog een foto te zien van de opstelling van de 3D printer.



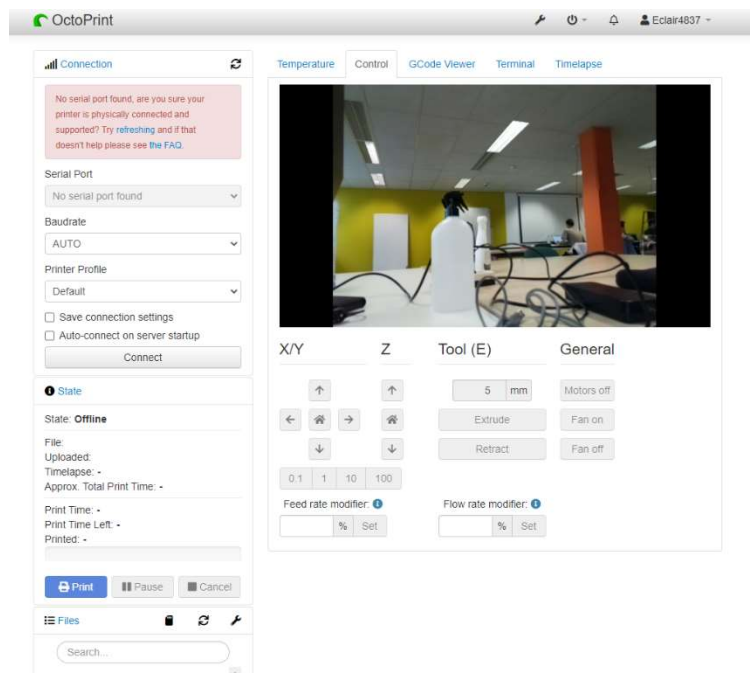
Afbeelding 1 Foto opstelling 3D printer

3 Results

3.1 Raspberry Pi verbinden met 3D printer

Om te beginnen moest de Raspberry Pi met de 3D printer verbonden worden. Dit gedeelte is opgesplitst in 2 delen. Eerst is er onderzocht geweest wat de beste optie was voor ons doeleinde. Al vlug werd Octoprint gevonden. Met Octoprint kan de printer volledig bestuurd worden, dit is gemaakt voor op een Raspberry Pi, heeft een volledige kant en klare webservice en kan een webcam feed weergeven. Octoprint bleek dus ideaal voor dit project.

Daarna kwam het testen van Octoprint, de installatie ging vlot met de "Raspberry Pi Imager". Er moest een wachtwoord, gebruikersnaam, wifi-naam of SSID en wifi-wachtwoord ingegeven worden. Na het booten van de image kan er naar <http://raspberrypi.local> gegaan worden of via een netwerktool het IP-adres achterhaald worden en zo er mee verbonden worden. Na het inloggen werd onderstaand beeld weergegeven.



Afbeelding 2 Octoprint webpagina

Dan moest enkel no de USB ingestoken worden tussen de printer en de Raspberry Pi en op connect geklikt worden op de webpagina.

3.2 PrusaSlicer

Om 3d-bestanden was PrusaSlicer de betere omdat deze gemaakt is voor het bepaald merk printers dat er gebruikt wordt op school. In de software kan ook het model printer dat gebruikt wordt gekozen worden zodat alle instellingen optimaal zijn. Met andere slicers ondervonden we ook dat de prints rechtsachter geprint werden. Met PrusaSlicer werd dit probleem niet meer ondervonden en werd een thumbnail toegevoegd zodat op de webpagina het onderdeel direct bekeken kan worden. Er is ook voor een tutorial gezorgd dat studenten geen problemen kunnen ondervinden.

3.3 Raspberry Pi tijdzone

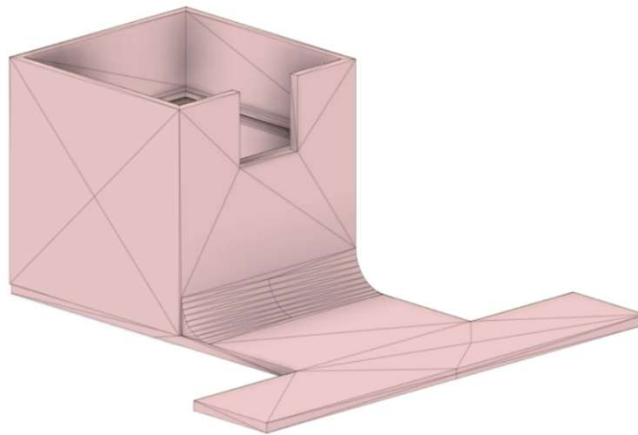
Doordat de tijd een aantal uur verkeerd stond liet de webpagina de verkeerde tijd zien wanneer een print klaar ging zijn. Om dit aan te passen moest er verbinding worden gemaakt met de Raspberry Pi via ssh. Daarna moesten er een aantal commando's ingegeven worden. Dan sprong er een wizard op en deze moest gevolgd worden om de tijd juist in te stellen.

3.4 Firmware updaten van 3d-printer

De printer firmware is geüpdatet omdat de printer elke keer bij het starten van een print de melding gaf dat er firmware update beschikbaar was. Een gemakkelijke manier was om op de webpagina een plugin te installeren. De plugin "FirmwareUpdater" zorgt ervoor dat een firmware update gemakkelijk en vlog geïnstalleerd kan worden. Eerst moesten een paar instellingen specifiek voor dat model printer ingesteld worden. Dan moest de firmware gedownload worden op de Prusa website. Dan moet het hex-bestand naar de plugin geüpload worden en op "Flash" gedrukt. Na een aantal minuten is dit gereed.

3.5 Bumper

Voor de 3D printer continu te laten printen moet er iets zijn dat het geprinte werkstuk van het printbed afhaalt. Het eerste gedacht was om de drager van de printkop de print ervan af te laten duwen. Dit bleek fout te lopen wanneer er lage prints moesten weggeduwd worden omdat deze tegen de extruder en de ventilator aankwamen. De oplossing hiervoor was een soort bumper die van de drager tot juist boven het printbed zou gaan. Deze beschermt hierdoor zowel de extruder als de ventilator tegen een lage print maar is ook hoog genoeg om niet tegen de print te komen tijdens het printproces. Zo ontstond de eerste versie van de bumper

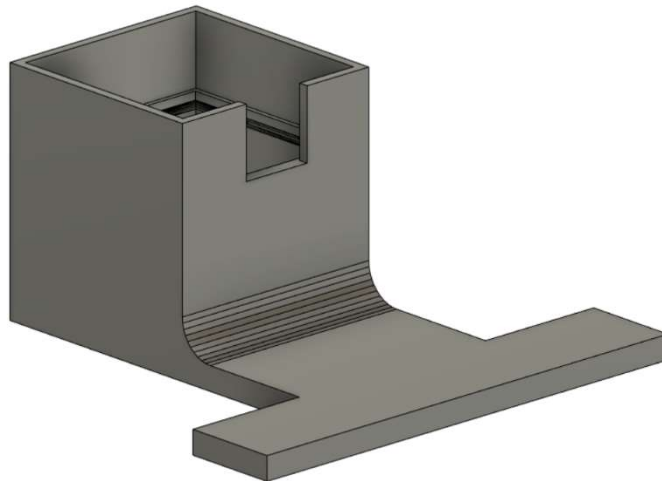


Figuur 1 Bumper versie 1

Maar tijdens het testen van deze bumper waren er enkele dingen opgevallen.

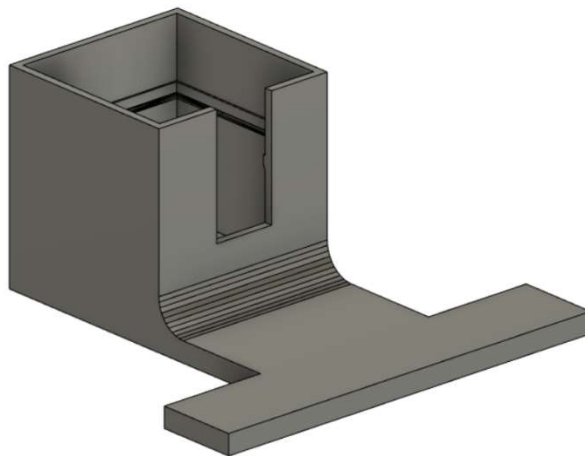
1. De lip die aan de binnenkant zit was te breed om over de stappenmotor te passen.
2. De onderkant van de bumper had maar een dikte van 2 a 3mm. Dit was genoeg om een kleine print van het bed af te duwen maar zou de sterkte niet hebben om een grotere print van het bed af te krijgen.

Met deze opmerkingen was de tweede versie van de bumper ontworpen.



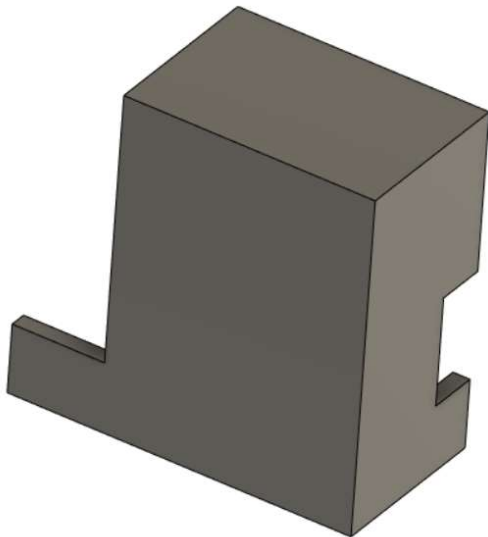
Figuur 2 Bumper versie 2

Tijdens het testen van de tweede versie waren er toch nog enkele punten die niet volledig in orde waren. Zo was er een insnijding in de bumper voorzien voor de bedrading van de stappenmotor. Deze had een lengte van 15mm maar het probleem was dat de insnijding een 10 – 15mm langer had moeten zijn. Met al deze opmerkingen was versie drie ontstaan.

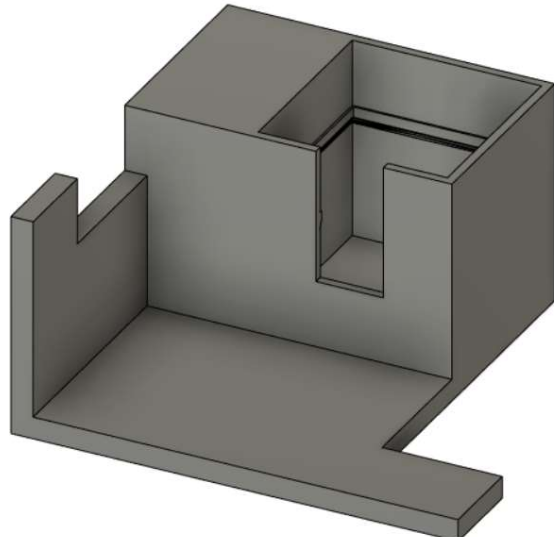


Figuur 3 Bumper versie 3

Deze had nu een insnijding van 30mm waardoor deze over de stappenmotor paste zonder dat de bedrading in de weg zat. Dit model heeft enkele weken op de printer gestaan tot dat er de waarneming kwam dat wanneer de printer zijn bed clearing ging doen deze zijwaarts ging bewegen. Hierdoor kon het mogelijk zijn dat er een print vast kwam te zitten tussen de bumper en de behuizing van de ventilator. De oplossing hiervoor was een tweede zijde maken aan de bumper en zo werd er versie vier ontworpen.



Figuur 5 Voorkant bumper versie 4

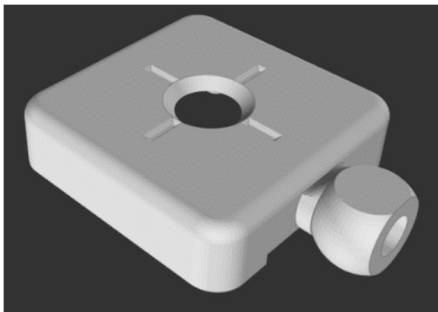


Figuur 4 Achterkant bumper versie 4

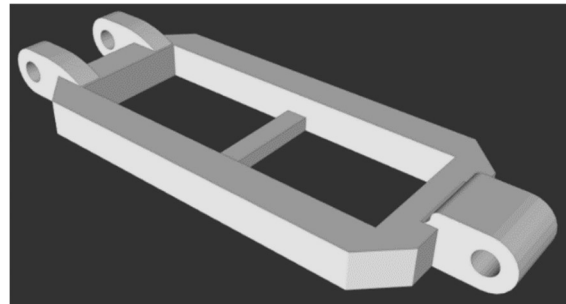
Versie vier van de bumper kon nu zowel voorwaarts als zijwaarts de print van het printbed afduwen.

3.6 Camera en camera staander

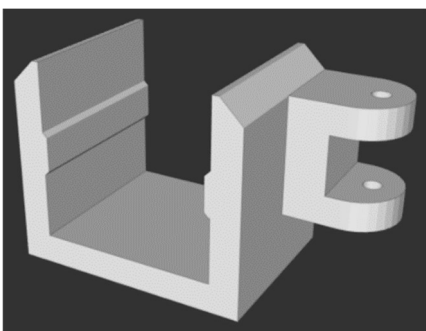
Op Octoprint is er een optie om live camera beelden te bekijken. Deze beelden worden voorzien via een camera die verbonden is met de Raspberry Pi. De camera die gebruikt wordt voor dit project is een *Raspberry Pi Camera Module 2*. Maar om de printer goed in beeld te krijgen is er een camera staander nodig. Aangezien het project rond een 3D printer draaide zou het niet meer dan logisch zijn dat deze ook zelf geprint zou worden. De camera bestaat uit enkele onderdelen: een beugel, een verlengstuk, een camera case en een drager voor de camera case.



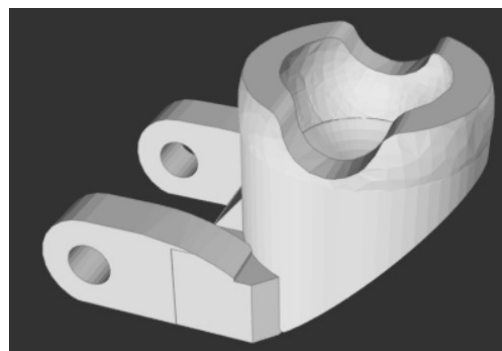
Drawing 1 Camera behuizing



Drawing 2 Verlengstuk



Drawing 4 Beugel



Drawing 3 Camera behuizing drager

Deze onderdelen waren gevonden op Thingiverse maar waren zelf geprint en geïnstalleerd. Bij het installeren waren er nog enkele bouten en moeren noodzakelijk voor alle onderdelen aan elkaar te monteren. Door het installeren van de camera staander op de printer was de standaard meegeleverde camerakabel niet lang genoeg voor de camera in alle posities te zetten. Hierdoor is er een kabel besteld geweest die 75cm lang was. Deze kabel en de bouten en moeren waren voorzien door de lectoren.



Afbeelding 3 Opstelling camera module

3.7 Toegangsgroepen maken

De printer zou als einddoel toegankelijk moeten zijn voor alle elektronica-ict studenten. Iedere student krijgt hiervoor zijn persoonlijke login. Deze studenten zullen dan maar een beperkte toegang krijgen tot Octoprint. Dit is mogelijk door de student een aparte toegangsgroep te geven. Deze groep is zo aangemaakt dat er enkel mee geprint kan worden en geen instellingen of dergelijke kunnen gewijzigd worden. De lectoren krijgen ook een persoonlijke login maar deze zullen in een aparte groep zitten zodat deze zowel kunnen printen als ook instellingen aanpassen.

3.8 Plugins

Octoprint kan gebruik maken van plugins. Zo gebruiken wij enkele plugins zoals:

3.8.1 Display layer progress

Deze zal de informatie / vooruitgang van de print ophalen en op de server laten zien. Deze informatie zal ook te zien zijn op het printerdisplay. DisplayLayerProgress zal ook samen werken met de volgende plugin.

3.8.2 Print time genius

De PrintTimeGenius zal de tijd berekenen die de printer nodig zal hebben om deze print te printen als ook de resterende tijd voor de print te voltooien. Deze functie gebeurt ook standaard door de printer zelf maar de berekeningen van PrintTimeGenius zijn toch een beetje beter en accurater. Deze met behulp van de DisplayLayerProgress op het printerdisplay te zien zijn

3.8.3 Filament manager

Deze manager helpt bij het beheren van de filamentspoelen. Een functie van FilamentManager is dat deze zal onthouden hoeveel gram filament er nog op een bepaalde spoel zit. Dit is handig wanneer er constant van kleur wordt gewisseld. Het is ook handig dat de manager een foutmelding zal geven wanneer er te weinig filament over is op de spoel voor de print volledig af te maken.

3.8.4 Octo text

OctoText zal meldingen geven bij de ingestelde vereisten. Deze meldingen gebeuren via email. Het enige wat nodig is voor OctoText is een emailadres met het bijhorende wachtwoord die de mails zal versturen. Zo kan octotext een mail bij volgende opties:

- Melding geven bij:
 - o Start van print
 - o Einde van print
 - o Bestand geüpload
 - o Pauze van de print
- Mail sturen wanneer er een error plaatsvindt

Vooraleer octotext een mail kan sturen zal er eerst nog een configuratie van de email moeten gebeuren. Het configureren gebeurt als volgt:

SMTP-gateway = smtp.office365.com

SMTP port = 587

Het emailadres dat gebruikt wordt voor octotext is octoprint@outlook.be

3.9 Continu printen

Alle printjobs worden in een queue geplaatst waardoor de printer deze achter elkaar kan printen. Wanneer de print voltooid is zal er doormiddel van een bedclearing script de print van het bed worden verwijderd en kan de volgende job beginnen. Hierdoor is het niet nodig om fysiek aanwezig te zijn bij de printer. Alle gebruikers voegen hun print toe aan de queue en kunnen volgen wanneer hun print start en eindigt.

<input type="checkbox"/>	Job?	Progress?	local
<input type="checkbox"/>	Job 3	1/1	
	moerbuis_0.2mm_PLA_MK3_12m....	1/1	
<input type="checkbox"/>		1/1	
	Cilinder_0.15mm_PLA_MK3S_6m....	1/1	
<input type="checkbox"/>	Job 1	1/1	
	Cilinder_0.15mm_PLA_MK3S_6m....	1/1	
<input type="checkbox"/>		3/3	
	Cilinder_0.15mm_PLA_MK3S_7m.g...		▲
<input type="checkbox"/>		1/1	
	Cilinder_0.15mm_PLA_MK3S_7m.g...		▲
<input type="checkbox"/>	Job 2	1/1	
	Cilinder_0.15mm_PLA_MK3S_7m.g...		▲
	Cilinder_0.15mm_PLA_MK3S_6m....	1/1	

6

Afbeelding 4 Queue continuous print

3.10 Raspberry Pi via ssd

De Raspberry Pi wordt geboot vanaf een SSD. De reden hiervoor is dat een SSD stabielere zijn als meeste micro-sd kaarten. Een bijkomend voordeel is dat een SSD sneller is dan een micro-sd. De OS van OctoPi is geflashed op de SSD en verbonden met de Raspberry Pi. Hierdoor kan de Raspberry Pi booten via SSD en is OctoPi toegankelijk voor gebruikers.

3.11 Bed clearing

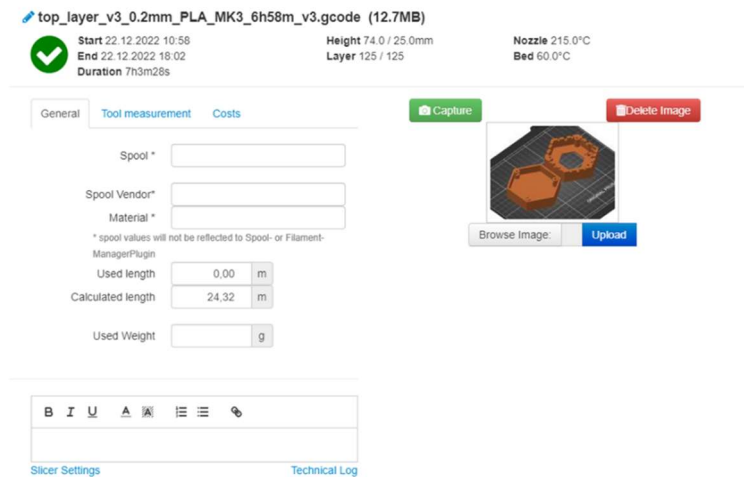
Door gebruik te maken van het bed clearing script kan de printer het printstuk van het bed verwijderen en klaarmaken voor de volgende print job. Nadat de print klaar is zal de gcode worden uitgevoerd. Deze code zorgt ervoor dat de nozzle in de linker achterhoek begint, de beginpositie van het bed is op het voorste maximale punt. Hierna zal het bed naar achter bewegen en zal de eerste sweep beginnen. Nadat de eerste sweep is geëindigd zal het de nozzle omhooggaan en het bed terug naar voren. De nozzle beweegt nu naar het midden van het bed en gaat naar beneden. Het bed gaat naar achter en start de tweede sweep. Hierna gebeurt hetzelfde als na sweep 1 en zal de nozzle naar de rechterkant van het bed bewegen. Het bed beweegt naar achter en start de derde sweep. Er wordt op verschillende plaatsen van het bed een sweep gedaan zodat er zekerheid is dat het bed leeg is in geval van meerdere stukken in dezelfde job. Als de printer het script heeft doorlopen is het bed klaar voor de volgende job te starten.

```
G1 Z 210 F3000 ; move up
G1 X0 Y200 Z 210 F3000 ; move back and sideways
G1 X0 Y200 Z 80 F3000 ; move down
G1 X0 Y000 Z 80 F3000 ; move forward
G1 X0 Y200 Z 80 F3000 ; move backwards
G1 X0 Y200 Z 1 F3000 ; move down
G1 X44 F3000; move in x for first sweep
G1 Y45 F3000; move forward for first sweep
G1 Z50 F3000; move up
G1 Y200 F3000; move back
G1 Z1 F3000; move down
G1 X116 F3000; move in x for first sweep
G1 Y45 F3000; move forward for second sweep
G1 Z50 F3000; move up
G1 Y200 F3000; move back
G1 Z1 F3000; move down
G1 X209 F3000; move in x for first sweep
G1 Y45 F3000; move forward for second sweep
G1 Z50 F3000; move up
G1 Y200 F3000; move back
G1 Z1 F3000; move down
```

Afbeelding 5 Bed clearing code

3.12 Log bestanden

Er worden logfiles bijgehouden na iedere print zodat er een overzicht is van iedere print en de gebruiker die deze heeft aangemaakt. Er wordt een korte beschrijving van de print bijgehouden zoals de hoogte en het aantal layers. De starttijd, de eindtijd en de printtijd is ook terug te vinden. Er wordt een geschatte lengte van het aantal filament dat gebruikt zal worden getoond en ook de werkelijke lengte die gebruikt is. Dit kan ook worden gebruikt om de kosten van de print te berekenen als er een prijs wordt meegegeven van de filament en andere kosten die zijn gebeurd. Als de print wordt geannuleerd of mislukt wordt dit ook opgeslagen zodat de gebruiker weet wat er gebeurd is met de print als de job niet succesvol was. Er wordt een screenshot toegevoegd om snel te kunnen zoeken welke print van de gebruiker is.



Afbeelding 6 Voorbeeld log bestand

3.13 Wan verbinding

De Raspberry Pi waar octopi op draait moet toegankelijk gemaakt worden voor mensen van buitenaf. Dit is zodat de gebruikers niet fysiek aanwezig moeten zijn voor bestanden naar octopi door te sturen. De oplossing hiervoor is om de Raspberry Pi te verbinden met de server die gebruikt wordt door het departement elektronica. Deze server zal dan via buitenaf beschikbaar zijn via volgende website: <http://octopi.pxl-ea-ict.be:24081/> deze zal gebruik maken van een WAN verbinding met als poort 24081 om verbinding te maken met de Raspberry Pi.

3.14 Script gebruikers

De bedoeling is dat studenten via de webpagina hun print gaan uploaden zodat deze in een queue komt. Maar het is ook de bedoeling dat dit per student ook allemaal kan bijgehouden worden. Hiervoor moest elke student een eigen gebruiker op de webpagina hebben. Om dit handmatig elk jaar opnieuw te doen is veel te veel werk. Het idee is dus om met een script alle gebruikers aan te maken. Om dit te doen moest er eerst een "API Key" gegenereerd worden, dit kan onder "Settings" en dan "Application Keys". Dit script bestaat uit enkele delen. Een gedeelte is om de mailadressen uit een CSV-bestand te halen zodat dit niet manueel ingevuld moet worden. Een ander gedeelte is voor elke gebruiker dat aangemaakt wordt moet een random sterk wachtwoord gegenereerd worden. Er is ook een gedeelte om het gedeelte voor de "@" van het mailadres als gebruikersnaam te gebruiken. Dan worden de gebruikers aangemaakt. En om af te sluiten wordt er een mail gestuurd naar dit mailadres met de gebruikersnaam en wachtwoord. Ook is er nog de mogelijkheid om gebruikers via een script te verwijderen, dit kan ook via het uitlezen van een CSV-bestand.

4 Discussion

Het project is tot een samenhangend geheel gekomen waarmee studenten en lectoren van afstand kunnen printen zonder manuele tussenkomst. De gebruikers krijgen hiervoor een persoonlijk account toegewezen waarmee ze op de server (octopi) kunnen printen. Voor het printen op de server is een tutorial gemaakt die uitleg geeft over welke software er nodig is en de bijhorende instellingen die nodig zijn voor de huidige printer. Verder zal er ook nog uitleg gegeven worden over hoe het printbestand in de wachtlijst gezet moet worden. En als slot nog een kleine hulplijst die helpt bij het oplossen van mogelijke fouten bij het printen.

5 Conclusion

De onderzoeksvraag was om een 3D printer van op afstand bruikbaar te maken voor alle studenten van PXL. Dit lijkt op eerste zicht te voldoen aan de opdracht die werd gegeven. Het is mogelijk om via een webapplicatie de printer te besturen en te printen. Doormiddel van een queue is het ook mogelijk om alle prints achtereenvolgend te printen.

Tijdens deze opdracht zijn er ook enkele problemen voorgekomen die werden opgemerkt tijdens het testen van de printer. Zo zijn er bijvoorbeeld enkele voorwaarden waaraan de print moet voldoen om een succesvol eindresultaat te bekomen van de print.

Een tweede probleem is dat als de print van het bed wordt verwijderd deze onder het bed valt. Dit is tijdelijk opgelost doormiddel van een papier over het gat te plaatsen en de print over het papier glijdt. Dit kan in de toekomst worden opgelost door de printer onder een hoek te plaatsen.

Kort samengevat is de opdracht succesvol uitgevoerd en is de printer beschikbaar voor gebruik.

6 Reference list

[1]

Watch It Print, “It’s Easy! Automated Continuous 3d Printing Que On My Ender 3,” *YouTube*. Mar. 22, 2021. [YouTube Video]. Available: <https://www.youtube.com/watch?v=NZgj8W5yrE>

[2]

michaelnew, “michaelnew/Octoprint-Print-Queue,” *GitHub*, Jun. 13, 2018.
<https://github.com/michaelnew/Octoprint-Print-Queue>

[3]

Z. Rashid, “Automating My 3D Printer - The Startup - Medium,” *Medium*, Mar. 03, 2021.
<https://medium.com/swlh/automating-my-3d-printer-81b4f653d4d0>

[4]

The Real World, “Prusa MK3S+ Raspberry Pi - Octoprint - Webcam Installation (Part 2 of our Prusa Series).,” *YouTube*. Jan. 23, 2022. [YouTube Video]. Available:
<https://www.youtube.com/watch?v=0FWOMdLVRjg>

[5]

“Continuous Print,” *Github.io*, 2023. <https://smartin015.github.io/continuousprint/> .

[6]

G. Häußge, “OctoPrint.org,” *OctoPrint.org*, 2022. <https://octoprint.org/> .

[7]

Thingiverse.com, “Thingiverse - Digital Designs for Physical Objects,” *@thingiverse*, 2023.
<https://www.thingiverse.com/>