Fontys Hogescholen Eindhoven
Technische informatica

3D-Led-Kubus

Gemaakt door:

Pieter Compen

AFSTUDEERVERSLAG VOOR FONTYS HOGESCHOOL ICT

Gegevens studenten:	
Naam + voorletters student	Compen P.P.J.
Studentnummer	2112456
Opleiding	Technische Informatica
Afstudeerperiode	29-08-2011 t/m 07-01-2012
Gegevens Bedrijf:	
Naam bedrijf/instelling:	TASS technology solutions
Plaats	Eindhoven
Gegevens bedrijfsbegeleider:	
Naam	Rop Pulles
Functie	People manager
Gegevens technisch begeleider:	
Naam	Bart Bilos
Functie	Embedded Software Engineer
Gegevens verslag:	
Title afstudeerverslag met subtitel	3D-led-kubus
Datum uitgifte afstudeerverslag	

Getekend voor gezien door bedrijfsbegeleider:

Datum:

De bedrijfsbegeleider,

Voorwoord

Ter afsluiting van mijn studie Technische Informatica aan Fontys hogescholen te Eindhoven heb ik een half jaar een afstudeerstage gedaan bij TASS te Eindhoven.

Bij de zoektocht naar een geschikte afstudeerplek had ik me voorgenomen om een opdracht te zoeken waarbij embedded hardware een rol ging spelen. Vanuit school had ik al meerdere lovende woorden gehoord over TASS. Dit bedrijf had meerdere opdrachten online staan waaruit gekozen kon worden, mijn intentie was om te beginnen aan een interactieve led-wall. Na een gesprek gehad te hebben met TASS bleek dat deze niet meer beschikbaar was, maar ze hadden wel een alternatief: een 3D-led-kubus. Dit leek mij ook een zeer leuke opdracht, zodoende heb ik deze aangenomen.

De opdracht bestond uit het ontwikkelen van de aansturingssoftware voor de kubus, een onderzoek naar een geschikte microprocessor en om na te denken over een constructie waarmee de kubus eenvoudig mee te vervoeren is.

De opdracht was nooit gerealiseerd zonder de hulp van een aantal medewerkers binnen TASS. Allereerst wil ik mijn bedrijfsbegeleider, Rop Pulles, bedanken voor zijn sturing en feedback op mijn werkzaamheden. Daarnaast wil ik mijn technisch begeleider, Bart Bilos, bedanken voor de feedback op alle hardware-vragen en daarnaast de realisatie van de hardware.

Ook wil ik Luc Compen, bedanken voor het beschikbaar stellen van zijn apparatuur en het maken van een positief-film voor de glazen-printplaat test.

Uiteraard wil ik mijn docentbegeleider, Eric Dortmans, bedanken voor de snelle feedback en voor alle antwoorden op de vele vragen die ik had.

Inhoudsopgave

Samenvatting

De afstudeerstage is gedaan bij TASS Technology solutions, een dienstverlener op het gebied van technische en embedded software. Oorspronkelijk een onderdeel van Philips en sinds 2007 een zelfstandige onderneming. De kernactiviteiten van het bedrijf zijn: Mobility & Automotive, Healthcare & Cure, Mechanics & Control, Consumer Lifestyle en Safety & Security.

Om meer naamsbekendheid te creëren bij bedrijven en studenten staat het bedrijf regelmatig op verschillende vakbeurzen. Om hier tussen de rest op te vallen staan er op de stands veelal demo's. TASS is altijd op zoek naar nieuwe innovatieve demo's en een led kubus is hier ook een voorbeeld van.

Voor deze afstudeerperiode waren al een aantal keren pogingen ondernomen om een werkende led kubus te maken. Nu was dit tot op heden nog niet goed gelukt, de eerste vraag die dan ook werd gesteld bij de aanvang van het project is: "Waarom is dit niet gelukt". Later bleek dat er verschillende zaken waren die er voor zorgde dat er nog geen werkend exemplaar aanwezig was: Een onstabiele kubus, veel kortsluiting bij de aansluiting van de leds, slechte keuze qua microcontroller en hardware die niet op tijd beschikbaar was.

Allereerst is er gekeken naar een goede keuze van microcontroller en welke eigenschappen deze nodig zou hebben. Hieruit is een ontwikkelbord uit voortgekomen. Met behulp van Linux is er begonnen aan de ontwikkeling van de software. Door een modulaire opzet en herkenbare softwarepatronen is de software goed te begrijpen voor mede-software engineers.

Tijdens de implementatie zijn er wat zaken naar voren gekomen die het proces een stuk hebben vertraagt, zo is er het niet goed kunnen aansturen van de PWM-module op de microcontroller. Wat een belangrijk stuk van de aansturing is. Dit zal worden opgelost door de PWM te laten generen met behulp van een simpelere microcontroller die wordt aangestuurd door het LPC2468 bord. Verder is tijdens het project de hardware wat laat geleverd zodat er behoorlijk laat pas kon gewerkt worden aan de drivers voor de led-aansturing.

Al met al is uiteindelijk een goede basis gevormd qua software voor de besturing van een led kubus. Er zal nog even gewerkt moeten worden aan de aansturing via de LPC2468 microcontroller. Verder is de opzet zo gemaakt dat het eenvoudig is om nieuwe functionaliteiten toe te voegen aan de kubus.

Summary

This report is about the graduate internship carried out at TASS Software Professionals in Eindhoven. TASS is a software company that originated from Philips. The focus of the company is in the area of technical and embedded software. The activities where TASS specializes in are: Mobility & Automotive, Healthcare & Cure, Mechanics & Conrol, Consumer Lifestyle and Safety & Security.

One of the recent project was a project in which a LED cube has been developed. The device can display animations and text in three dimensions. A special software tool, the LED cube editor application, lets users control the LED cube.

There are, however, multiple problems with the Cube. For instance the cube itself isn't stable and breaks down very easy, another problem is the lack of good software that controls the cube internally.

The main goal of this graduate internship was to develop a completely new 3D LED cube. Where the problems mentioned earlier will be solved. Besides that, the cube will be controlled by the LED cube editor application and eventually will operate in a stand-alone mode.

First of all a suitable micro-controller has been selected from a wide range of different platforms. A development board from Embedded Artists has been chosen. The development board runs a version of Linux called μCLinux which has been designed especially for micro-controllers. With a modular setup of the written program and usage of software patterns the complete solution is easy understandable for other software engineers.

With the development of embedded software, there where some problems. For example the controlling the led-driver chips with a pulse width modulation signal wasn't possible in combination with the data signals. Which resulted in a non-functional driver. The solution was to add an external micro-controller with a good working pwm signal. Secondly the hardware, build by a employee of TASS, was delivered with a long delay.

Above all the result is a structured design of the software. At the end there are some modifications were needed for a complete cube. The connection between the LED cube editor application and the cube itself has been established. The software is specifically designed so it is fairly easy to add extra functionalities to the program.

Verklarende woordenlijst

3D	Het begrip driedimensionaal of 3D duidt aan dat iets drie meetkundige dimensies heeft: diepte, breedte en hoogte. Voorbeelden van drie-dimensionale dingen zijn een bol, een piramide of een ruimtelijk object zoals een schoenendoos.
FPS	Het aantal beelden per seconde (Engels: frames per second, FPS), is een maat die aangeeft hoe snel een apparaat beelden (frames) weergeeft of anderzijds verwerkt.
BPS	Het aantal bits dat per seconde wordt verstuurd over een communicatielijn.
LAN	local area network (lokaal thuisnetwerk); twee of meer computers die rechtstreeks, of via een gedeeld medium met elkaar verbonden zijn.
Led	Een led is een elektronische component, een halfgeleidercomponent die licht uitzendt als er een elektrische stroom in de doorlaatrichting doorheen wordt gestuurd.
PMP	Project Management Plan, het initiele document wat beschrijft hoe het project wordt aangepakt.
PWM	Pulsbreedtemodulatie (In het Engels Pulse-width Modulation, afgekort tot PWM) is een modulatietechniek die wordt gebruikt als vorm van elektrische voeding en als een manier van digitale informatieoverdracht.
USB	USB (Universal Serial Bus of Universele Serile Bus) is een standaard voor de aansluiting van randapparatuur bij computers.
uv	Ultraviolet (afgekort uv) is elektromagnetische straling net buiten het deel van het spectrum dat met het menselijk oog waarneembaar is.

1 Inleiding

In de huidige periode is driedimensionaal (3D) een heel hip onderwerp, in de bioscoop, op de tv, spelcomputers, allemaal kunnen ze 3D-beelden weergeven. Met een range van verschillende technieken lijkt het alsof de kijker midden in de film zit. Veelal wordt er gebruik gemaakt van een speciale bril waarbij elk oog een ander beeld voorgeschoteld krijgt.

Figuur 1.1: Led

Ook met behulp van meerdere light emitting diodes (Led) kan er een 3D-beeld gecreëerd worden, de zogenaamde led-kubus. Met de kubus is het mogelijk om eenvoudige 3D-animaties weer te geven. Omdat het redelijk goedkoop te maken is worden deze kubussen vooral gemaakt door hobbyisten, ook zijn deze verkrijgbaar als bouwpakket en zelfs als compleet product.

Zoals veel bedrijven staat TASS ook op beurzen om zo naamsbekendheid te creëren bij potentiële klanten en eventuele nieuwe werknemers. Om toch het verschil te maken op een beurs wil het bedrijf graag nieuwe gadgets om zo de aandacht op zich te krijgen. Een led-kubus is hier een voorbeeld van.

Figuur 1.2: Led Kubus

Er zijn verschillende soorten leds, zo zijn er leds die maar 1 kleur kunnen weergeven, meerdere kleuren en zelfs zowat alle, voor de mens, zichtbare kleuren. Met de laatste soort is het de bedoeling een kubus te maken.

2 Het bedrijf

2.1 Geschiedenis

TASS Technology Solutions, of in het kort TASS, is een dienstverlener op het gebied van technische en embedded software. Het bedrijf is in 1978 opgericht door Philips als Special Products Group.
TASS is sinds 2007 onderdeel van Total Specific Solutions. Tot op heden is het bedrijf één van de top 3 spelers in de benelux voor technische software dienstverlening, met ruim 200 medewerkers in Nederland en België.

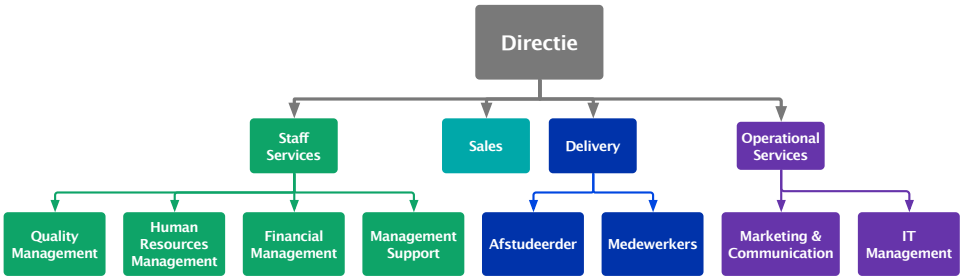
2.2 Kernactiviteiten

De kernactiviteiten van TASS zijn: Mobility & Automotive, Healthcare & Cure, Mechanics & Control, Consumer Lifestyle en Safety & Security.

Figuur 2.1: TASS hoofdkantoor

2.3 Organisatie

De focus bij TASS ligt vooral bij de ontwikkeling van technische en embedded software voor externe klanten. Grotendeels zijn de medewerkers gedetacheerd bij bedrijven zoals Philips, ASML, Bosch, Océ en TNO. Elke medewerker van TASS krijgt bij aanvang een people manager toegewezen, de contactpersoon tussen TASS en het externe bedrijf waar hij of zij is gedetacheerd.



Figuur 2.2: Organogram van de organisatie

In figuur 2.2 staat de organisatie van TASS Nederland schematisch weergegeven. Bovenaan in het organogram staat de directie, onder leiding van Edwin Manten (algemeen directeur) en Bert van Elburg (Commercieel directeur). Daaronder is TASS globaal gezien op te delen in vier afdelingen:

Staff Services dit is de overkoepelde afdeling voor *Quality Management, Human Resources, Financial Management en Management Support*.

Sales de afdeling *Sales* houdt zich onder andere bezig met het werven van nieuwe klanten en het onderhouden van bestaande klantrelaties. Daarnaast worden recente marktontwikkelingen gevolgd om hier zo mogelijk op te anticiperen.

Delivery alle gedetacheerde medewerkers en people managers van TASS vallen onder deze afdeling. Ook studenten die stage lopen bij TASS, vallen onder Delivery; de afstudeerstage vond dus plaats binnen deze afdeling.

Operational Services bestaat uit twee subafdelingen: *Marketing & Communication* en *IT Management*. Deze afdelingen voorzien achtereenvolgens in de interne en externe communicatie en in de voorzieningen op het gebied van ICT (te denken valt aan computers en netwerkbeheer).

3 De opdracht

3.1 Beginsituatie

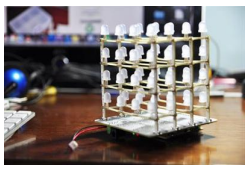
TASS is al langer bezig geweest met het realiseren naar een led-kubus, het is helaas nog niet gelukt om daadwerkelijk ook een goede kubus te fabriceren. Bij de vorige pogingen tot het maken van de kubus is er veel kennis opgebouwd, beginnend met deze kennis zal de nieuwe kubus worden opgebouwd.
TASS heeft zelf al een *led-cube-editor* gemaakt in een vorig afstudeerproject. Met de editor is het mogelijk om grafisch eenvoudig animaties te maken voor een led-kubus.

3.2 Gewenste eindsituatie

Omdat er al meerdere projecten waren geweest met betrekking tot de led-kubus, was in het al snel duidelijk wat de klant, TASS, precies wilt. Daarbij waren nog wel wat vragen over hoe de kubus nu precies in elkaar moest worden gezet. Er was al geprobeerd met koperdraad en betonijzer, bijde pogingen zijn niet geslaagd. De kubus was of niet te transporteren of zag er niet uitnodigend uit.
Eén van de vraagstukken van TASS was hoe de kubus zo te maken was dat deze stevig genoeg was om te transporteren naar de verschillende beurzen.
Ook was er geen werkende embedded software aanwezig die de kubus kan aansturen, dit is voor de opdracht één van de belangrijkste producten die zal worden afgeleverd.

3.2.1 Stabiele kubus

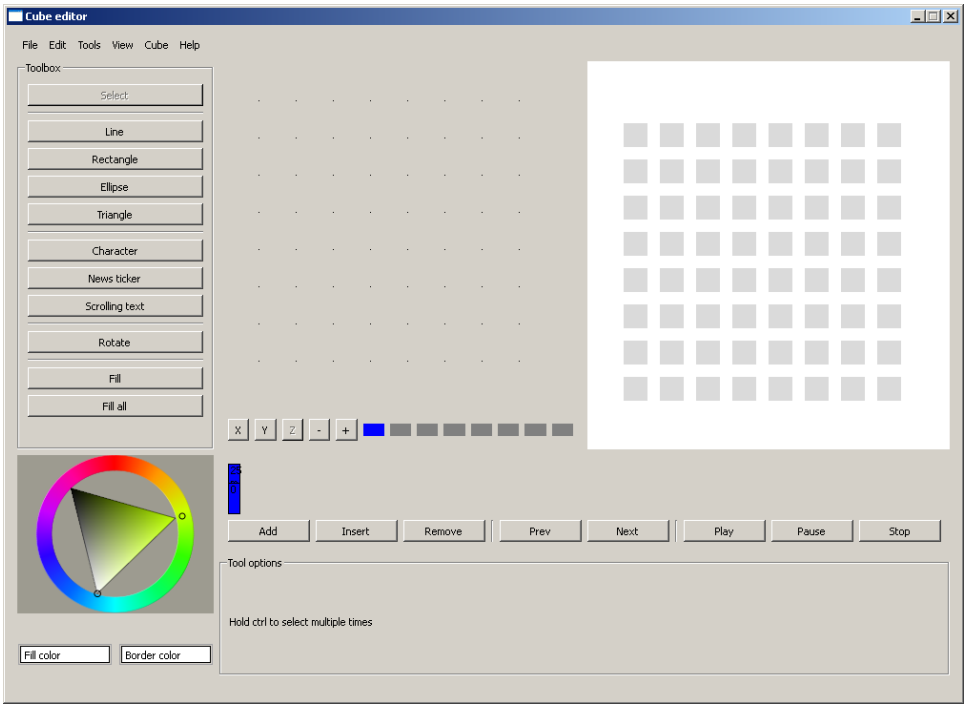
Als eerste idee was om gebruik te maken van printplaten om zo de kubus in elkaar te zetten, hiermee zou de kubus zeer stabiel worden, ook het vervangen van eventuele defecte ledjes zou hiermee zeer eenvoudig zijn. Ook is het uiterlijk van de kubus redelijk aantrekkelijk om naar te kijken.



Figuur 3.1: Voorstel constructie kubus

3.2.2 Connectiviteit

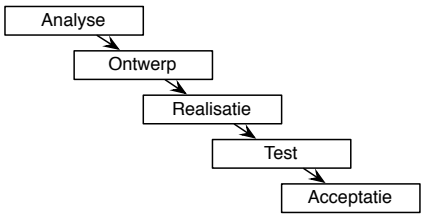
Verder is er de requirement dat de kubus kan communiceren met externe apparaten, denk hierbij aan een pc en sensoren.
Zoals eerder vermeld is er een pc applicatie speciaal ontwikkeld voor de kubus, deze zal moeten worden herschreven om comptabiliteit met de nieuwe kubus te kunnen garanderen.



Figuur 3.2: Cube Editor software van TASS

3.3 Projectmethodiek

De Gebruikte projectmethodiek tijdens de afstudeerstage is de *Aorta-lifecycle*. Deze projectmethodiek is gebaseerd op het waterval-model maar kent meer terugkoppelingsmomenten dan het klassieke waterval-model. De redenen voor de keuze voor deze methodiek is dat de specificaties niet continue aan verandering onderhevig zijn. Het is eenvoudig om met de methodiek te werken.



Figuur 3.3: Het Aorta-lifecycle model

Eén van de nadelen is wel dat het voor kan komen dat er relatief veel werk wordt gespendeerd aan de eerste fases en dat daardoor het risico ontstaat dat er weinig tijd overblijft voor het realiseren en testen.

Oriëntatiefase Deze fase is voornamelijk bedoeld om kennis te maken met het bedrijf en de opdracht duidelijk te krijgen aan de hand van de wensen van de klant/gebruiker en eventueel inlezen als er al documentatie aanwezig is.

Planningsfase In de planningsfase wordt er een Project Management Plan (PMP) opgesteld en een planning gemaakt.

Analysefase In deze fase worden de specificaties opgesteld voor de microcontroller met behulp van een onderzoek en zal moeten worden voldaan aan de wensen van de klant.

Ontwerpfase Deze fase is bedoeld voor het ontwerp van alle zaken die aan bod komen voor het project, zo zal het duidelijk moeten worden hoe de software in elkaar gaat zitten en eventuele hardware zaken moeten duidelijk zijn zodat deze goed overeenstemmen met de software.

Realisatiefase In de realisatiefase zal vooral worden geprogrammeerd en de hardware zal moeten worden geassembleerd, eventueel zal er bestaande software aangepast worden.

Testfase Deze fase is bedoeld voor het uitvoeren van verschillende tests, om zo de kwaliteit van de software te kunnen bewaren en eventuele fouten weg te werken die nog niet zijn voorgekomen tijdens de realisatiefase.

Acceptatiefase De laatste fase zal het product worden opgeleverd en zal deze worden gepresenteerd aan de klant.

3.4 Opdrachtomschrijving

De taak in het project is om software te maken die de aangeleverde hardware (Ledkubus) aan te sturen. Verder moet er uitgezocht worden welke microcontroller het geheel kan aansturen. Bij deze omschrijving komen er een aantal deelvragen naar voren zoals: ”welke aansluitingen zijn er nodig?”, ”wat is het budget?”, ”Hoe zal de aansturing tot zijn werk gaan?”, ”Hoe is het mogelijk om data, gemaakt met de led cube editor, uit te lezen en weer te geven met de kubus-software?”.

Software De kubus moet extern en via sensoren te besturen zijn. Verder is er een ledkubus editor aanwezig bij TASS, met deze software moet het mogelijk zijn om patronen in de kubus te programmeren.

Hardware De hardware zal worden geleverd en ontworpen door de technisch begeleider, Bart Bilos. De hardware zal zo worden opgezet dat deze aan te sluiten is op de meeste microcontrollers. Welke microcontroller compatibel zijn met de hardware zal moeten worden uitgezocht. Ook de algemene werking van de hardware zal nog moeten worden onderzocht. Dit omdat in vorige projecten andere hardware is gebruikt.

4 Onderzoek

4.1 Inleiding

Omdat er al redelijk veel informatie aanwezig is bij TASS, zal er eerst onderzocht moeten worden waarom de vorige projecten niet slaagden.
De eerste versie van het project werkten men met het idee om met koperdraden de led's aan elkaar te solderen. Al snel was het probleem dat er veel kortsluiting was in de constructie van de kubus, ook de gebruikte hardware had last van storingen.
Bij het volgende project werd er gebruik gemaakt van betonijzer, wat op zich wel een stevige constructie is, het nodigt niet echt uit voor het oog. Ook was de hardware niet op tijd aanwezig zodoende dat er geen eindproduct was om te demonstreren, maar wel veel informatie waarom dit ook niet goed werkte.

Het belangrijkste van het onderzoek is om een goede keuze te maken naar een geschikte microcontroller. Omdat het ontwerp van de aansturings-module bij aanvang van het project begon al aanwezig was, moet hier een goede keuze naar gemaakt worden.

Zo is bij de keuze belangrijk hoe snel de doorvoersnelheid van de chip moet zijn, de hoeveelheid aansluit-pinnen er aanwezig moeten zijn om de led's uiteindelijk aan te sturen en ook is de connectiviteit met een ander apparaat belangrijk. Er is een windows-programma aanwezig waarmee het mogelijk is om animaties voor een led-kubus te maken, het is dan ook belangrijk dat de te maken kubus hiermee overweg kan.

4.2 Opbouw kubus

De eerdere versies van de kubus waren niet goed functioneel met name doordat het constructief niet goed in elkaar zat. Hierdoor komt al snel de vraag: "hoe de kubus in elkaar te zetten zodat deze ook goed vervoerbaar is?"naar voren. Eén van de eerste ideeën waren om met behulp van printplaatstrips de leds te monteren, hierdoor blijven de leds altijd op hun plaats. Nu is een van de nadelen bij het gebruik van de ledstrips dat niet uit alle hoeken de leds zichtbaar zijn. Het is dus de vraag of er een mogelijkheid is om deze ledstrips doorzichtig te maken.

4.2.1 Glazen printplaat

Na wat zoekwerk op internet kwam het er op neer dat het niet eenvoudig is om een bedrijf te vinden die goede doorzichtige printplaten kan fabriceren. Bij de zoektocht naar doorzichtige printplaten. Kwam echter naar voren dat er ook een andere manier is om de printplaten te maken[8]. Namelijk met glas in plaats van plastic. Het is een eenvoudig proces: met ultraviolet (uv) lijm koperfolie op glas plakken en daarna via de gebruikelijke manier etsen. Omdat het niet bekend was wat voor effect het etsmiddel heeft op het glas en lijm is eerst een proefexemplaar gemaakt op een klein stuk glas.

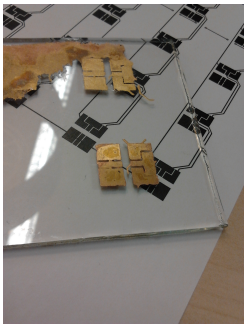
Eerste test

De test is elektrisch gezien mislukt maar qua techniek goed gelukt. Zo heeft etsmiddel geen tot zeer weinig effect op het gebruikte glas (vensterglas). En het koper blijft goed zitten op de glasplaat. Wel is het wenselijk om gebruik te maken van zo dun mogelijke laag koperfolie, dit zal de etsperiode aanzienlijk verlagen. Ook is de

gebruikte toner-transfer techniek niet eenvoudig toe te passen op de glazen platen en zal in een volgende test worden vervangen met een lichtgevoelige laag.

Tweede test

Bij de tweede test heb ik de hulp ingeroepen van een professionele offset drukkerij, deze hebben namelijk materiaal aanwezig waarbij het mogelijk is om zeer grote printplaten te belichten en te etsen.



Figuur 4.1: Eerste test glazen printplaat

Omdat het onduidelijk was welke belichtingstijd er nodig is om de plaat te belichten, is er eerste een proefplaatje gemaakt van 5×5 cm. Deze wordt eerst beplakt met het koper, via de aanwezige lichtbak droogt de lijm. Hierna kan er de lichtgevoeligelaag over het koper worden gespoten (in een donkere kamer). Na het belichten wordt het geheel ontwikkeld waarna de printplaat geëtst kan worden in een mix van H_2O_2 en HCl.

De testprint is goed gelukt, hierna is over gegaan naar een grotere plaat: 30×30cm. Al snel kwamen de eerste moelijkheden naar voren. Zo is het niet eenvoudig om in een donkere kamer gelijkmatig een lichtgevoelige laag op de plaat te spuiten. Bij de gedeeltes waar er iets meer over was gespoten moest het ook langer belicht worden. Omdat het niet goed mogelijk is om de plaat gedeeltelijk te belichten zijn er meerdere pogingen gedaan om de laag opnieuw aan te brengen op de glazen plaat. Na tig pogingen is uiteindelijk de handdoek in de ring gegooid en is uiteindelijk besloten om de kubus toch fabriceren met de voorgestelde ledstrips.

4.3 Keuze microcontroller

Om de kubus aan te sturen zal er gebruik worden gemaakt van een microcontroller. Om een goede basis te hebben zijn er zogenaamde ontwikkelborden op de markt. Hiermee is het mogelijk om zeer snel tot een eindproduct te komen. Omdat dit de basis zal worden van het eindproduct zal eerst goed moeten worden onderzocht wat de beste keuze is. Er zijn een aantal zaken waar het ontwikkelbord aan moet voldoen. Zo moet er een aansluiting zijn voor externe opslag-apperatuur, mogelijkheden tot het aansluiten van externe sensoren en moet er tegen een redelijke prijs een ontwikkelomgeving aanwezig zijn.
Verder zal het mogelijk moeten zijn dat de aangeleverde hardware zal functioneren met de microcontroller

4.3.1 Inputs

Om uiteindelijk animaties weer te kunnen geven op de kubus zal er een vorm van connectiviteit naar de microcontroller moeten zijn. De minimale snelheid waarmee dit moet gebeuren is te bepalen met een eenvoudige berekening 4.1.

$$ledaantal \times bits \times fps = bps \tag{4.1}$$

Allereerst zal moeten worden bepaald hoeveel stappen in grijswaarde de leds zullen weergeven (*bits*). De aangeleverde hardware maakt gebruik van, in serie geschakelde, TLC5940 chips. Per chip zijn er 16 Pulse Width Modulation (PWM)

uitgangen, deze kunnen per kanaal 12 bits aan verschillende niveaus aan.

Voor de weergave van verschillende kleuren wordt er gebruik gemaakt van RGB[1]-leds, voor de kubus wil dit zeggen dat er 3 kanalen naar elk led gaat. Wat uitkomt op 36 bits aan verschillende kleuren per led. Vergelijkend met een normaal computerscherm, die in theorie alle zichtbare kleuren zou weergeven, is dit behoorlijk. Daarom zal de input van de kleurenwaarde gereduceert worden naar 8 bit per kleur.

De kubus zal in totaal $(8 \times 8 \times 8) = 512$ leds gaan aansturen waarbij elke led 3 verschillende kleuren bezit. Dit zal resulteren in een totaal van $(512 \times 3) = 1536$ aansluitpunten (*ledaantal*).
Het belangrijkste van de microcontroller is, dat deze de led-aansturingsmodule kan besturen en via externe apparaten kan worden aangestuurd.

Als laatste zal moeten worden bepaald wat het maximum aan frames per seconde (fps) zal zijn voor de animaties weer te geven de kubus. Kijkend naar de traditionele film waarbij er 24 frames per seconde wordt getoond is gekozen voor maximaal 30 frames per seconde. Dit geeft wat meer zekerheid dat de frames soepel worden weergegeven.

Nu alle variabelen bekend zijn kan berekening 4.1 worden ingevuld:

$$1536 \times 8 \times 30 = 368640 \tag{4.2}$$

Het is nu dus van belang dat de gekozen input meer dan 368640 bits per seconde kan doorgeven. Voor de connectie tussen de pc en microcontroller zijn er verschillende mogelijkheden, de meest voorkomende zijn: de seriële poort, Universal Serial Bus (USB), Local Area Network (LAN).

Seriële poort

Op de huidige pc's zijn tal van manieren om connectie te creëren met de buitenwereld. Een simpele bus is de seriële poort, deze wordt voornamelijk nog gebruikt bij kleine projecten. Dit met name doordat het het zeer eenvoudig aan te sluiten is op externe apparaten. Het grote nadeel van de poort is dat het een relatief lage doorvoersnelheid heeft: maximaal 115200 bits per seconde (*bit/s*). Als we deze gegevens invullen in berekening 4.1 komt men op het volgende resultaat:

$$\frac{115200}{1536 \times 8} = 9,38 \tag{4.3}$$

Met de seriële poort kan dus theoretisch gezien maximaal 9,38 frames per seconde overgestuurd worden. Wat voor deze toepassing te weinig is.

USB 1.0

Een andere aansluitmogelijkheid is via USB. Dit is op het moment dé standaard als het gaat om externe apperatuur aan te sluiten op de pc. Ook heeft deze bus een hogere doorvoersnelheid als de seriële poort: 1500000 bit/s. (1,5 Mbit/s)

$$\frac{1500000}{1536 \times 8} = 122,07 \tag{4.4}$$

Volgens berekening 4.4 zal met behulp van deze poort ongeveer 122 frames kunnen worden verzonden. Wat voor de toepassing ruim voldoende is.

[1] Red Green Blue

LAN

Steeds meer wordt er naast USB ook gebruik gemaakt van een netwerkaansluiting. De doorvoersnelheid van de meeste producten met een netwerkaansluiting is 100000000 bit/s (100 Mbit/s).

$$\frac{100000000}{1536 \times 8} = 8138,02 \tag{4.5}$$

Theoretisch gezien zal dit dus resulteren op 8138,02 frames per seconde. Waar wel rekening mee moet gehouden worden is dat er nogal wat overhead zit op de tcp/ip pakketten, maar dit heeft niet zo'n grote invloed.

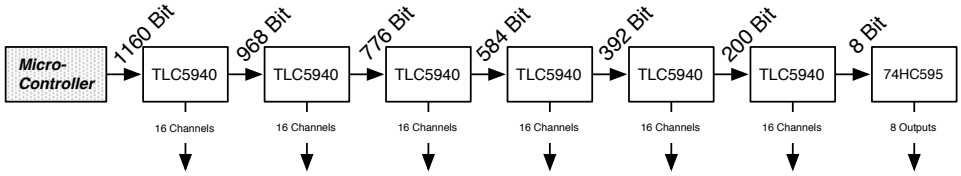
Resultaat

Voor de juiste doorvoersnelheid zal dus het ontwikkelbord een aansluiting nodig hebben met USB of LAN. In figuur E.1 is Serieel en USB uitgewerkt in een grafiek, LAN is niet opgenomen in deze grafiek omdat deze buiten het bereik valt.

4.3.2 Outputs

Buiten dat de microcontroller signalen moet ontvangen, zal deze ook signalen moeten zenden en wel naar het controllerbord voor de led's. Voor de keuze van een microcontroller zal eerst duidelijk moeten zijn hoeveel outputs er aanwezig moeten zijn.

Het controllerbord bestaat uit serie geschakelde led-drivers en aan het einde van de keten een shiftregister voor de horizontale assen (zie figuur 4.2).



Figuur 4.2: Flow van shiftregisters en aantal bits wat doorgestuurd wordt

Om te bepalen hoeveel aansluitingen nodig zijn zal eerst gekeken worden naar de aansluit-pinnen van de led-driver. Informatie van de fabrikant wijst aan dat er minimaal 5 aansluitingen nodig zijn om deze chip aan te sturen. Verder zijn deze 5 aansluitingen gegroepeerd in 2 verschillende funties. Eén groep zorgt voor de informatie van de grijswaarde, de andere groep zorgt dat de leds daadwerkelijk aan gaan (met behulp van een PWM-signaal).

4.3.3 Development Boards

Nu de hoofdzaken duidelijk zijn waar de microprocessor aan moet voldoen, zal er vergeleken moeten worden welk development board het meeste geschikt is om de kubus aan te sturen.

Voor de selectie van de verschillende borden moet er rekening worden gehouden met de verschillende eigenschappen van de verschillende chips. Eén van de voorwaarden is dat een aansluiting met een pc is die een doorvoersnelheid heeft van ten minste 552960 bit/s (zie berekening 4.2). Verder is het belangrijk dat er gekeken wordt naar de aanwezigheid van een SD-kaart, het aantal te gebruiken I/O aansluitingen, de CPU, de programmeertaal, kosten en de aanwezigheid van een

ontwikkelomgeving. In bijlage B is een overzicht van alle onderzochte development boards met daarbij alle concrete technische specificaties en conclusies.

Development Board	↗ 0,5 MBit/s	↗ 4 I/O pinnen	Extern geheugen	ADC	LAN	USB	Levertijd (in werkdagen)	Prijs
Arduino Ethernet	-	x	x	x	x	-	2	€ 47,95
Olimex PIC32-MX460	x	x	x	x	-	x	2	€ 514,81
FEZ Panda II	x	x	x	x	-	x	2	€ 40,41
AVR-USB-STK	x	x	x	-	-	x	2	€ 29,00
Embedded Artists LPC2468	x	x	x	x	x	x	0	€ 169,00

Tabel 4.1: Overzicht onderzochte development boards

4.3.4 Conclusie

Tijdens de zoektocht naar een geschikte microcontroller zijn er een hele range aan ontwikkelborden de revue gepasseerd. Uiteindelijk is de keuze gevallen op het LPC2468 bord. Wat doorslaggevend is gebleken is de directe beschikbaarheid, de snelle processor, de vele aansluit-mogelijkheden, de mogelijkheid tot het gebruik van C++ in combinatie met Linux en de vele beschikbare resources en support van Embedded Artists.

4.4 LPC2468

4.4.1 Inleiding

Nadat de keuze gevallen is voor het ontwikkelbord, zal er gekeken worden naar de ontwikkelomgeving voor het bord. Bij het LPC-bord is een VMware-image meegeleverd met daarop een aangepaste versie van de GCC-compiler[2]. Ook is er support vanuit Embedded Artists in de vorm van een uitgebreid document waarin stap voor stap staat uitgelegd hoe men een werkend geheel krijgt[7].

4.4.2 Software

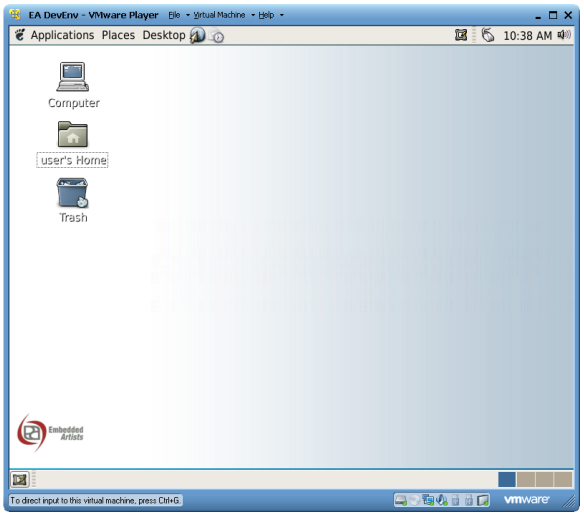
Het belangrijkste van de opdracht is werkende aansturingssoftware op te leveren. Tijdens eerdere pogingen zijn er verschillende stukken code geschreven en opgeleverd. Na wat onderzoek is besloten om deze niet verder te gebruiken en opnieuw het geheel te maken. De keuze tot het LPC ontwikkelbord geeft de mogelijkheid tot het gebruik van embedded Linux.

In deze ontwikkelomgeving is een aangepaste versie van μCLinux meegeleverd. Op de ontwikkelomgeving is achteraf eclipse geïnstalleerd waardoor ontwikkeling van programma's eenvoudiger is.

μCLinux

De gebruikte variant van Linux is eigenlijk een aangepaste versie. Deze is speciaal bedoeld om te werken op kleine embedded systemen die niet alle functionaliteiten

[2]Programma om code om te zetten naar uitvoerbare machine-taal.



Figuur 4.3: Aangeleverde ontwikkelomgeving van Embedded Artists

hebben van de gebruikelijke architecturen zoals de *x*86 architectuur.

Connectie met pc Om verbinding te maken met de hardware wordt er standaard gebruik gemaakt van een seriële telnet verbinding. Hiermee is het mogelijk om commando's uit te voeren en eventueel bestanden over te zenden vanuit de pc naar het bord en andersom. Omdat serieel al is afgeschoten in het vorige onderzoek zal gebruik worden gemaakt van de aanwezige netwerkverbinding. Hierbij zijn twee keuzes te maken voor bestanden over te zenden: FTP[3] of SAMBA[4].

De eerste keuze is het SMB-protocol, hiermee kunnen er zeer eenvoudig bestanden overgestuurd worden met het Windows besturingssysteem. Bij het compileren van de SMB implementatie komen er zoveel fouten naar voren dat het te lang gaat duren om het werkend te krijgen. hierbij is dan ook gekozen om gebruik te maken van FTP, de software is daarbij ook een substantieel kleiner qua dataruimte.

Geheugenmanagement Hoewel er gebruikt gemaakt kan worden van een standaard besturingssysteem zijn er een aantal restricties. Het is bijvoorbeeld niet mogelijk om aan geheugenmanagement (MMU[5]) te doen. Dit wil zeggen dat een programma bij het geheugen kan dat eigenlijk niet bedoeld is voor het programma zelf. Zodoende is er dus ook geen vangnet als het programma crashed en kan daardoor het gehele systeem laten vastlopen.
Natuurlijk kan deze situatie ook voordelen voordeling zijn. Het maken van zogenaamde besturingsdrivers, de verbinding tussen de software en hardware, is niet nodig. Een applicatie kan zodoende direct de hardware aansturen zonder dat de gebruiker extra software moet installeren.

Drivers Doordat er de mogelijkheid is om via een applicatie direct met de hardware te communiceren is er gekozen om geen drivers te schrijven. Dit geeft minder overhead omdat er niet met de Linux kernel moet worden gecommuniceerd. Bij de keuze van een ander platform met MMU zal er later wel een driver moeten wor-

[3] **F**ile **T**ransfer **P**rotocol

[4] Samba is een opensource implementatie van het SMB-protocol, wat staat voor **S**erver **M**essage **B**lock

[5] De **M**emory **M**anagement **U**nit is een hardware-component in de computer dat gebruikt wordt voor de runtime-mapping van virtuele naar fysieke geheugenadressen.

den geschreven. Door een modulaire opzet van het te schrijven programma zal dit weinig werk vragen.

Ontwikkeling Door het gebruik van Linux, is het mogelijk om een keuze te maken tussen verschillende programmeertalen. Buiten C is het bijvoorbeeld ook mogelijk om te programmeren in C++. Bij grotere projecten is het overzichtelijker om daarbij gebruik te maken van een object georiënteerde taal. In samenwerking met een UML-ontwerp is de structuur van het programma sneller te begrijpen voor externe personen.

5 Ontwerp

5.1 inleiding

In het volgende hoofdstuk staat beschreven hoe het uiteindelijke ontwerp er uitziet en hoe het is geïmplementeerd in het systeem. Het is verdeeld in twee categorieën: Hardware en Software. Omdat de kernactiviteiten niet lagen in de hardware, staat hierin alleen beschreven wat het ontwerp was voor het idee van de glazen printplaten, hiervoor is namelijk een printplaatontwerp voor gemaakt.

In het software gedeelte staat het softwareontwerp beschreven en hoe uiteindelijk de implementatie eruit ziet.

5.2 hardware

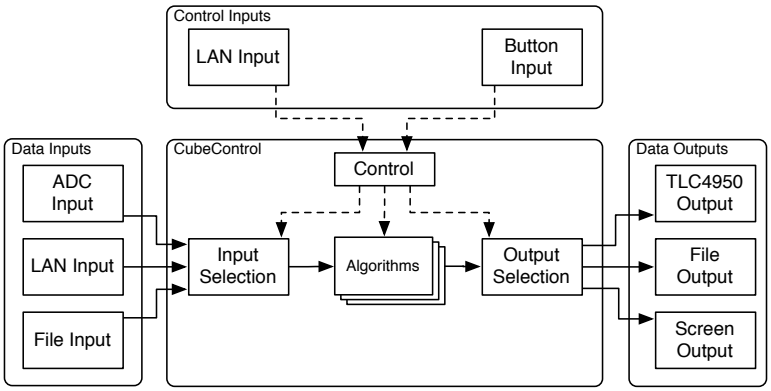
Tijdens de afstudeerperiode is ook een hardware-ontwerp gemaakt voor de glazen printplaten. Deze is gemaakt met behulp van het programma *Eagle CAD*. De glazen printplaat is 30×30 cm groot, per plaat zijn er aansluitingen voor de rijen en 24 aansluiting voor elke led aan te sturen. In totaal zal dit uitkomen op 200 verbindingen naar de kubus toe. De grootste uitdaging voor het ontwerp is dat het niet mogelijk is om te etsen op 2 kanten, dit omdat er geen glasboor bestaat waarbij de diameter kleiner is dan 3 mm. Zodoende is het niet mogelijk om de ene zijde te verbinden met de andere. Met deze gegevens is uiteindelijk een design uit voort gekomen, deze is te bekijken in bijlage D.

5.3 Software

De software, *CubeControl*, moet zo ontworpen worden dat het eenvoudig te begrijpen is en zodanig in elkaar zitten dat extra functionaliteiten eenvoudig toe te voegen zijn. Om dit te realiseren is gekozen om gebruik te maken van Software Patterns[2]. Door het ontbreken van een MMU zal proefsgewijs uitgevonden worden wat niet en wat wel kan.

Als er gekeken wordt naar het hoogste niveau wat het programma eigenlijk moet doen komt men op drie verschillende processen: Input → Calculatie → Output. Om hieruit een concreet software patroon te halen, is gekozen voor bij elk proces het *Strategy Pattern* toe te passen. Met dit patroon is het mogelijk om per proces verschillende ”strategieën” toe te passen.

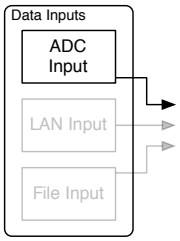
Nu moet het ook mogelijk zijn als het programma eenmaal gestart is dat de flow (Input → Calculatie → Output) real-time kan worden aangepast. Het programma zal dus extern signalen op moeten vangen, hierbij zal een tweede thread moeten worden opgestart die ”luistert” naar de externe signalen. Door het ontbreken van een MMU is het niet mogelijk om gebruik te maken van het traditionele Shared Memory en zal daardoor vervangen gaan worden door het *Singleton Pattern*. Een versimpelde weergave van het geheel is weergegeven in 5.1, een uitgebreider UML-diagram is te vinden in bijlage C.



Figuur 5.1: Opbouw en flow

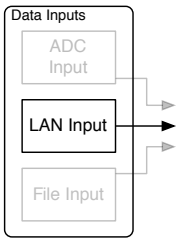
5.3.1 Input-modules

ADC



De eerste module is de aansturing van de Analog to Digital Coverter (ADC). Met de verkregen input is het mogelijk om interactie met mens en kubus te creëren. De interactie kan bijvoorbeeld gemaakt worden door middel van meerdere afstandssensoren op de kubus te monteren zodat de positie van de hand van een persoon te bepalen. Door verschillende posities zouden verschillende patronen kunnen worden weergegeven op de kubus.

LAN



Als tweede module is er de LAN-module, hierbij is het mogelijk om via het Local Area Network (LAN) data-pakketen over te sturen. Een extern apparaat kan de kubus direct aansturen. Om zo snel mogelijk data over te sturen van het externe apparaat naar de kubus is gekozen om het UDP-protocol te gaan gebruiken. Voor de datatoevoer is gekozen voor poort *30000*. Het UDP-datapakket ziet er volgt uit:

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12
Data	*Z*	*Y*	*R*1	*G*1	*B*1	*R*2	*G*2	*B*2	*R*3	*G*3	*B*3	*R*4	*G*4

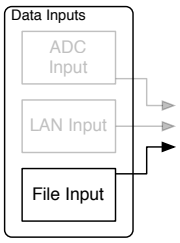
Byte	13	14	15	16	17	18	19	20	21	22	23	24	25
Data	*B*4	*R*5	*G*5	*B*5	*R*6	*G*6	*B*6	*R*7	*G*7	*B*7	*R*8	*G*8	*B*8

Tabel 5.1: Layout UDP-datapakket

De keuze voor UDP is gemaakt omdat dit protocol minder overhead heeft dan TCP-pakketen. Dit komt vooral doordat bij UDP geen terugkoppeling is met de client. Een nadeel van deze constructie is dat het niet controleerbaar is of de verzonden pakketten daadwerkelijk zijn aankomen bij de server/client.

Het UDP-protocol word dan ook vooral gebruikt bij streaming media zoals video's en muziek. Bij deze vorm van communicatie is het ook niet zo snel van belang dat er een aantal beelden niet door gestuurd worden.

File



Om vooraf bepaalde patronen weer te geven kunnen er verschillende bestanden worden ingeladen die data bevat waarbij de af te spelen patronen zijn opgeslagen. Deze patronen worden gemaakt met behulp van de 3d-editor, gemaakt in een vorig project bij TASS.
De inhoud van het bestand bestaat uit een header en datastream. In de header staan gegevens van de instellingen van de cube-editor, denk hierbij aan het aantal frames er zijn weggeschreven en voor welke grote kubus het bestand bedoeld is. De datastream bestaat uit x-aantal frames waarbij elke kleur opgeslagen is in 1 byte, oftewel 255 verschillende grijswaarden.

5.3.2 Algorithmes

Direct

Het direct-algorithme maakt een exacte kopie van de input naar de ouput. Het is dan ook van belang dat de input de juiste data op de goede manier doorgeeft.

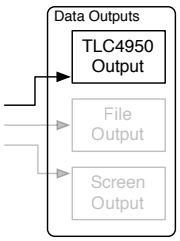
ADC

Het ADC-algorithme zorgt ervoor dat de kubus volledig verlicht wordt met het gebruik van 1 ADC input, in dit geval is dat potmeter AD0.0 op het LPC-bord.

Game Om via de ADC-input een spel te spelen is er het game-algorithme. Het algorithme berekend de positie van de spelers via lijnsensoren. Via de binnenkomde waardes kan eenvoudig verschillende posities worden berekend.

5.3.3 Outputs

TLC5940



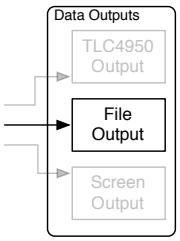
De kubus wordt aangestuurd door middel van zes schuifregisters die speciaal bedoeld zijn om leds aan te sturen. het TLC5940-output module zorgt ervoor dat de pakketten op de juiste manier in de schuif-registers worden ”ingeshift” en stuurt daarbij ook een 74HC595 aan voor de verticale lijnen.

De besturing van de chips gebeurd nu in het programma, deze kan namelijk direct de hardware aansturen. Omdat er gebruik wordt gemaakt van shift-registers is het niet nodig om voor elke chip een apart klok en data lijn aan te sluiten. Het is dus zonder enige moeite mogelijk om de kubus uit te vergroten, wel moet er rekening mee gehouden worden dat het shiften van de data wel langer kan gaan duren als er meer chips in een rij zijn geplaatst.

Voor de aansturing van de TLC5940 chip zijn er minimaal vier verschillende signalen nodig: Data, Clock, PWM-clock en Latch. Hierbij wordt de data via de combinatie Clock, data en latchsignaal doorgestuurd. Verder is er een PWM-signaal nodig om de grijswaarde van de leds aan te sturen. Per chip moet er $\frac{16chan.\times 12bit}{8} = 192$ bytes aan gegevens worden verstuurd, deze data moet verstuurd zijn voordat de 4096 klokslagen over het PWM-signaal zijn gegaan. Gebeurd dit niet dan zal de vorige waarde die in het geheugen stond worden gebruikt.

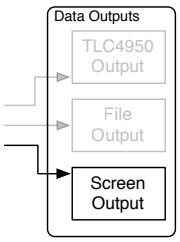
File

Het is mogelijk om de verstuurde input op te slaan in een bestand. File-output schrijft alle data in *output.cub*. omdat het een stream is die opgeslagen wordt moet men opletten dat het gebruikte geheugen niet vol raakt. De gegenereerde bestanden zijn tevens af te spelen met behulp van de File-input module.



Screen

De screen-output is bedoeld om de pakketten de debuggen die door de pipeline heen gaan. Het is daarmee ook een eenvoudige manier om te kunnen kijken of de algorithmes en inputs goed functioneren. Een van de nadelen van het gebruik van de screen-output is dat deze met een behoorlijke vertraging oploopt. Zodoende is het niet mogelijk om de kort timing uit te testen. In het huidige ontwerp worden alle frames weergegeven, de minimale tijd waarbij een frame wordt weergegeven zal bij het schrijven naar het scherm oplopen naar meer dan twee seconden per frame.



5.3.4 Control Inputs

De control inputs kunnen de inputs en outputs van *CubeControl* bepalen. Dit kan door middel van twee verschillende inputs:

LAN

Ook zoals bij de Lan input van de data kan via poort 30001 de verschillende statussen worden veranderd. Dit kan door middel van de volgende datapakket:

Byte	0	1
Data	*ID*	*Value*

Tabel 5.2: Layout UDP-besturingspakket

Hierbij kan het *ID* vier verschillende waarde bevatten waarmee de kubus te besturen is. *Value* is hierbij de waarde die in byte 2 mee gegeven wordt. In de gevallen dat deze niet gebruikt wordt zal deze dan ook genegeerd worden door *CubeControl*:

1: Volgende modus
2: Stel output-selectie met waarde *Value* in
3: Stel input-selectie met waarde *Value* in
4: Laadt alle externe bestanden in map */mnt/anim* opnieuw in

Buttons

Buiten de mogelijkheid tot de besturing van de kubus via LAN, is het ook mogelijk om zonder externe apperatuur de interactiviteiten van de kubus aan te passen. Dit gebeurd door middel van een drukknop die verbonden is aan het LPC-bord. Naar gelang van de verschillende actieve inputs zijn er verschillende statussen.

6 Implementatie

6.1 Bootloader

Om het geheel op te starten wordt er gebruik gemaakt van een bootloader. Dit maakt het eenvoudiger om nieuwe kernels te laden. Er is gebruik gemaakt van een standaard bootloader geleverd door Embedded Artists (ontwikkelaar van het LPC2468-bord), waarbij de niet gebruikte onderdelen zijn uitgehaald om zo het geheel zo snel mogelijk op te starten.

6.2 μClinux

Na de bootloader wordt μClinux uitgevoerd, deze keuze is gemaakt doordat er al veel kennis aanwezig was van embedded linux en zodat er snel kon worden begonnen aan het ontwikkelen van de software. Een ander voordeel is dat er gebruik gemaakt kan worden van standaard C++, hierbij is een architectuur ontworpen die zo modulair en herbruikbaar mogelijk is.

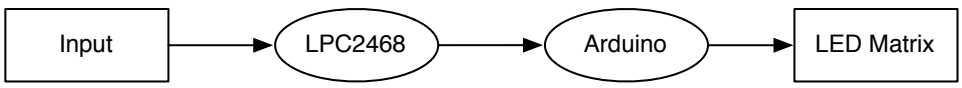
6.3 LPC2468

6.3.1 ADC

Om afstandssensoren uit te kunnen lezen zal er gebruik worden gemaakt van een ADC. Als voorbeeld levert Embedded Artists een klein aantal drivers mee, een van deze voorbeelden was een driver die de ADC van de LPC kan uitlezen. De driver gebruiken is redelijk eenvoudig, men leest het bestand in van de desbetreffende ADC en krijgt een getal terug. Bij het schrijven van de ADC-module in C++ voor CubeControl, kwam de fout naar voren dat alleen het 1ste getal gelezen werd. Bijvoorbeeld als de ADC de waarde 28 doorgeeft, las het programma alleen 2 in. Als men de ANSI-C implementatie gebruikt voor het uitlezen van bestanden gaat dit wel goed, er is dan ook voor gekozen om in het C++ project het inlees-gedeelte te vervangen door C-code.

6.3.2 PWM

Eén onderdeel, waar tijdens de opdracht behoorlijk wat moeite heeft veroorzaakt is, de PWM op de LPC. Het PWM-signaal wordt gebruikt om de Leds op een constante intensiteit te laten branden (Zie 4.3.2). Dit in combinatie met μCLinux leverde veel kernel-panics op, het was daarom zeer moeilijk te achterhalen wat de fout was. Ook debuggen met JTAG gaf weinig uitsluitsel. Zodoende moest er gezocht worden naar een (tijdelijke) oplossing. Uiteindelijk is besloten om er een AVR microcontroller tussen het LPC-bord en de led-aansturingsmodule te plaatsen. Een nadeel is wel dat de snelheid van 30 frames per seconde niet gehaald zal worden, dit doordat het aangesloten wordt met behulp van de seriële poort.



Figuur 6.1: Testopstelling met Arduino

6.3.3 Timing

Een belangrijk element in de weergave van animaties is timing. In de POSIX-standaard zijn verschillende methodes gedefinieerd hoe dit te realiseren is. Tijdens de opdracht is naar voren gekomen dat de gebruikte μCLinux niet echt meer up-to-date was. Ook de support voor de software is minimaal.

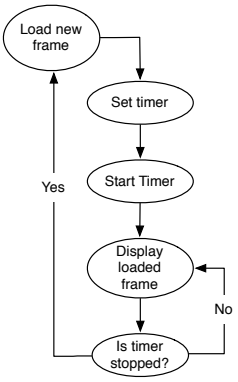
Omdat het nu eenmaal eenvoudiger is om zelf een timer te maken dan het geheel te updaten is hiervoor gekozen. Met de implementatie kan 1 hardware-timer aangestuurd worden. Deze kan in 2 verschillende modi opereren: Blocking en non-blocking.

Blocking

Bij blocking zal het programma wachten totdat de hardware-timer aangeeft klaar te zijn. Het wachten gebeurdt door middel van een while-loop, dit brengt met zich mee dat de cpu-load tijdens deze loop maximaal verhoogd zal worden. Het is dan ook aan te raden om in de toekomst een onderzoek te doen hoe het mogelijk is om deze te koppelen aan unix-signals [1], zodoende de cpu-load omlaag gaat.

Non-Blocking

De tweede modus is non-blocking. Dit wil zeggen dat een timer wordt geinitialiseerd waarna deze gestart kan worden. Het programma kan daarna zelf pollen of de timer is verlopen.
Tussen het checken door kan het programma andere zaken afhandelen om zo efficienter taken te verwerken.
De gehele timing-implementatie is speciaal gemaakt voor de LPC-architectuur. In andere omgevingen zal de implementatie gebruik kunnen maken van de standaard posix-timers.
Voor de weergave van de animaties met de kubus is gekozen voor het gebruik van de non-blocking timers. Bij elke nieuwe frame wordt de timer geinitialiseerd en ingesteld op de bepaalde tijd die bij de frame hoort. Daarna wordt de data weggeschreven naar de output. Hierna zal worden gekeken of de timer is verlopen, zo niet, dan zal hetzelfde frame naar de output worden geschreven.



Figuur 6.2: Timer CubeControl

6.4 Cube Editor

Om animaties te kunnen maken is er een Cube Editor gemaakt door TASS. Het programma is in staat om bestanden op te slaan en via de seriële poort data over te zenden naar de kubus-hardware. Nu wordt er in de nieuwe versie van de kubus geen gebruik gemaakt van seriële input. Om toch geen dubbel werk te creëren is het mogelijk om de opgeslagen bestanden te uploaden naar de kubus via FTP.
Er is getracht om ook via een Samba[2]-share dit te realiseren alleen werkte dit niet naar behoren op de kubus-hardware. Door het uitlezen van de bestanden is het mogelijk om zonder aanpassingen aan het programma de animaties weer te geven op de kubus.

[1] http://linux.die.net/man/7/signal

[2] Samba is een opensource implementatie van het SMB-protocol, hiermee is het mogelijk om bestanden uit te wisselen via het netwerk

7 Conclusies en aanbevelingen

De afstudeeropdracht die is uitgevoerd bestond vooral uit het ontwikkelen van de software. Er is ook gekeken naar de hardware, alleen had dit een lagere prioriteit. Er waren al meerdere pogingen gedaan om een werkende kubus te maken, maar deze zijn niet goed geslaagd. Eén van de redenen is dat er geen goede hardware aanwezig was. Tijdens deze afstudeerperiode is gezocht naar een ontwikkelbord waar zo min mogelijk aan toegevoegd diende te worden voor de aansturing. Verder is er nog een aansturingsbord gemaakt voor de aansturing van de leds, deze is door een medewerker van TASS gemaakt (Bart Bilos). Nu is het zo dat de hardware werd gemaakt in de vrije tijd van Bart en was tijdens de periode bezig met het verbeteren/vervangen van zijn infrastructuur wat zorgde voor vertraging in de oplevering van de hardware.

Kijkend naar het proces zijn er een aantal zaken aan te bevelen. Allereerst zou het handig zijn dat de hardware al aanwezig is voordat de opdracht aanvangt of een kant en klare led kubus te kopen. Verder zal er nog gekeken moeten worden naar de PWM van het LPC-bord, omdat het geheel modulair en op POSIX-besturingssystemen draait zal het niet moeilijk zijn om het geheel te porten naar een ander ontwikkelbord met betere ondersteuning waar bijvoorbeeld een goed werkend Real-time besturingssysteem op draait. Zodoende dat de PWM uitgang synchroon loopt met de data input van de led-drivers.

Evaluatie

Het half jaar afstuderen bij TASS is mij goed bevallen. Ik heb veel geleerd op het gebied van software design en de implementatie hiervan. De uitgevoerde werkzaamheden sluiten ook goed aan bij mijn verwachtingen.

In deze periode heb ik veel van de geleerde zaken kunnen toepassen in een professionele werkomgeving. Zo is er een vorm van een PID gemaakt, bij TASS heet dit een Project Management Plan. Ook is er een requirements document opgesteld oftewel het software requirements document. Door de voorbereiding waren deze documenten redelijk snel gemaakt.

Verder heb ik heb veel geleerd met betrekking tot programmeren in C++ en C, met name de mogelijkheden in een embedded omgeving. Door gebruik te maken van een besturingssysteem heb ik de geleerde software patronen bij Fontys kunnen toepassen in een embedded omgeving. Dit was een mooie leidraad wat er allemaal geprogrammeerd moest worden. Ook heb ik veel geleerd over hardware ontwerp en de mogelijkheden met de speciale led-drivers. Wat ook vrij nieuw voor mij was, was om in user-space hardware aan te sturen. Bij het programmeren van de hardware-aansturing, dat vaak het gehele systeem op tilt liet slaan, heb ik geleerd hoe belangrijk het is om stabiele code te schrijven, vooral met betrekking tot pointers en geheugen management. Dit ging vooral door veel te proberen en veel te lezen[1]. Doordat het gehele systeem vaak crashte door slechte code is de testfase en de realisatiefase zowat tegelijk gedaan.

Een van de zaken waar ik zelf niet echt tevreden over was, is het niet voor elkaar krijgen van een synchroon PWM signaal uit het ontwikkelbord. Verder is het jammer dat er geen prefab ledkubus aanwezig was. Dit zou de ontwikkeling een stuk eenvoudiger maken met betrekking tot de besturingssoftware.

Ik vond het het een heel leuke afstudeerperiode, ik heb veel geleerd, veel ervaring op gedaan en gewerkt in een leuke werksfeer waarbij ik veel nieuwe mensen heb leren kennen.

Literatuurlijst

[1] Brian W. Kernighan & Dennis M. Ritchie, *The C Programming Language, Second Edition*. Prentice Hall, Inc., 1988.

[2] Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.

[3] Grady Booch, James Rumbaugh, Ivar Jacobson, *Unified Modeling Language User Guide, 2nd Edition*. Addison-Wesley Professional, 2005.

[4] D. Jeff Dionne, *Supporting Linux Without an MMU*. Class #407. http://www.scribd.com/doc/41888500/Linux-Without-MMU

[5] Durrant, Michael and Leslie Of Lineo, Michael , *How uClinux provides MMU-less processors with an alternative*. EE Times, 2002. http://www.eetimes.com/electronics-news/4134390/How-uClinux-provides-MMU-less-processors-with-an-alternative

[6] Embedded Artists, *LPC2468 Developer's Kit User's Guide*.

[7] Embedded Artists, *uClinux Getting Started Book*.

[8] Kevin Dady, *Glass PCBs*. Hack a day, 2011. http://hackaday.com/2011/09/12/glass-pcbs/

[9] NXP B.V., *LPC24XX User manual*. 2009. http://ics.nxp.com/support/documents/microcontrollers/pdf/user.manual.lpc24xx.pdf

Appendices

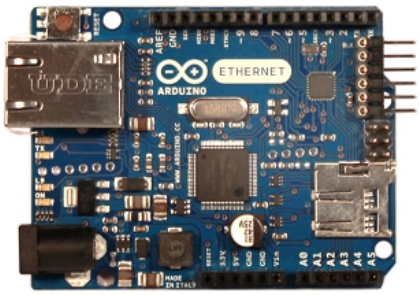
A Project Management Plan

B Development boards

B.1 Arduino Ethernet

B.1.1 Inleiding

De Arduino Ethernet is een standaard arduino waarbij de USB-aansluiting is vervangen door een ethernet-poort. Theoretisch gezien zou deze sneller data kunnen ontvangen en zenden.



Figuur B.1: Arduino Ethernet

B.1.2 Technische Specificaties

Eigenschappen

CPU	Atmega328
CPU speed	16Mhz
MIPS	16
I/O aansluitingen	14
SPI poorten	1

Analoge inputs	6
Ethernet	0,4...25 MBit
Programmeertaal	C
μSD	

Checklist

Connectie met pc > 0,5 MBit/s	×
> 4 I/O pinnen	√
Extern geheugen	√

Kosten

Ontwikkelbord	€	52,00
Ontwikkeltools	€	0,00
Totaal	€	52,00

B.1.3 Conclusie

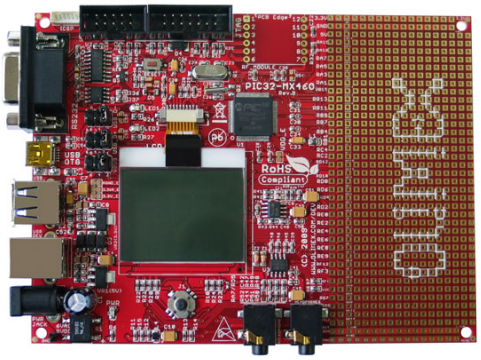
Het Arduino Ethernet bord is ideaal voor het gebruik in de kubus, het vergt wel wat programmeerwerk voor de LED-editor. Het connectiviteitsgedeelte van de software moet herschreven worden zodat deze via internet verbinding maakt. Over de snelheid zelf is weinig te vinden, de gebruikte ethernet-chip zou 25Mb aankunnen maar op verschillende fora word melding gemaakt van maximaal 500Kb[1] [2], wat waarschijnlijk komt door de trage processor en de overhead van TCP/IP. Door deze onduidelijkheid is het niet verstandig om dit bord te gebruiken met de kubus.

[1] http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1287138565/7#7
[2] http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1288488013

B.2 Olimex PIC32-MX460

B.2.1 Inleiding

De Olimex PIC32-MX460 is een ontwikkelbord wat gebruikt maakt van een PIC32 microprocessor. Deze processor heeft een snelheid van 80Mhz welke ongeveer één instructie per klokslag kan afhandelen. Ook heeft de processor een directe USB aansluiting, zodat er geen FT232-chip nodig is voor de communicatie.



Figuur B.2: PIC32-MX460

B.2.2 Technische Specificaties

Eigenschappen

CPU	PIC32MX460F512L	Analoge inputs	2
CPU speed	80Mhz	SPI poorten	2
MIPS	40	USB	O/H/D
I/O aansluitingen	70	Programmeertaal	C
SPI poorten	2	μSD	

Checklist

Connectie met pc > 0,5 MBit/s	√
> 4 I/O pinnen	√
Extern geheugen	√

Kosten

Ontwikkelbord	€	47,54
Ontwikkeltools	€	467,81
Totaal	€	514,81

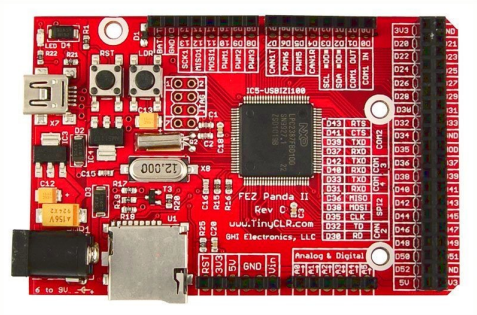
B.2.3 Conclusie

Het bord heeft alle eigenschappen die nodig zijn voor de aansturing van de kubus. De audio ingang kan gebruikt worden voor geluidsdetectie (weergave van FFT-analyse). Ook kan het beeldscherm handig zijn om informatie te laten zien over de huidige mode-selectie (aansluiting op pc, interactieve modus) of geheugengebruik ed. Het grote nadeel is de hoge aanschafkosten van de ontwikkeltools. Met de totale prijs van €514,81 is de aanschaf van het geheel niet aantrekkelijk.

B.3 FEZ Panda II

B.3.1 Inleiding

FEZ Panda II is een ontwikkelbord welke gebruik maakt van het .NET Micro framework, dit wil zeggen dat het geprogrammeerd kan worden in C#. Met Visual Studio kunnen de programmas worden geupload op het bord.



Figuur B.3: Fez Panda II

B.3.2 Technische Specificaties

Eigenschappen

CPU	NXP LPC2387	Analoge inputs	6
CPU speed	72Mhz	USB	O/H/D
MIPS	*Onbekend*	Programmeertaal	C#
I/O aansluitingen	54	μSD	
SPI poorten	2		

Checklist

Connectie met pc > 0,5 MBit/s	√
> 4 I/O pinnen	√
Extern geheugen	√

Kosten

Ontwikkelbord	€	40,41
Ontwikkeltools	€	0,00
Totaal	€	40,41

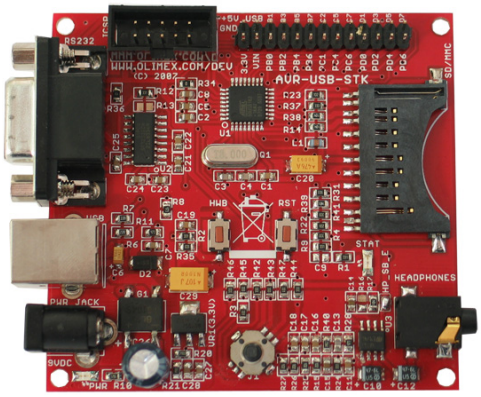
B.3.3 Conclusie

Het bord is ideaal voor de gebruikers die gewend zijn om in C# te programmeren, ook het gebruik van Visual Studio kan een voordeel opleveren. Verder kan de ontwikkeling alleen via het Windows besturingssysteem gedaan worden.

B.4 AVR-USB-STK

B.4.1 Inleiding

De AVR-USB-STK is een ontwikkelbord gemaakt door Olimex, wat gebruikt maakt van een AT90USB162. Het bord is te programmeren met een AVRISP mkII, deze is al aanwezig bij TASS.



Figuur B.4: AVR-USB-STK

B.4.2 Technische Specificaties

Eigenschappen

CPU	AT90USB162	Analoge inputs	0
CPU speed	8Mhz	USB	Device
MIPS	8	Programmeertaal	C
I/O aansluitingen	22	SD	
SPI poorten	1		

Checklist

Connectie met pc > 0,5 MBit/s	√
> 4 I/O pinnen	√
Extern geheugen	√

Kosten

Ontwikkelbord	€	29,00
Ontwikkeltools	€	0,00
Totaal	€	29,00

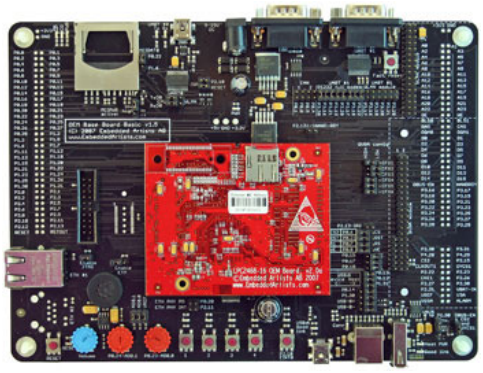
B.4.3 Conclusie

Het ontwikkelbord heeft de goede eigenschappen om gebruikt te worden in de kubus. Ook is het eenvoudig te programmeren en zijn deze spullen aanwezig bij TASS. De aanwezigheid van een SD-kaart aansluiting is ook zeer positief te noemen. Wat nog verder uitgezocht moet worden is de mogelijkheid tot het delen van de SPI-poort met de SD-kaart en de TLC5940 chips.
Verder zijn er geen analoge ingangen aanwezig, dit is verder niet echt een probleem maar dat zou opgelost kunnen worden door een externe ADC.

B.5 Embedded Artists LPC2468

B.5.1 Inleiding

De keuze voor het LPC2468 is omdat deze tijdens het onderzoek naar voren kwam bij TASS, deze was gebruikt bij een vorig project wat niet afgerond is en was afgesloten. Met het bord is het mogelijk om makkelijk zeer veel functies toe te voegen. Dit met name doordat er embedded linux op het systeem kan draaien. De software kan hierdoor gestructureerder opgezet worden.



Figuur B.5: LPC2468

B.5.2 Technische Specificaties

Eigenschappen

CPU	NXP LPC2468	Analoge inputs	8
CPU speed	72Mhz	USB	O/H/D
MIPS	80	Programmeertaal	C/C++
I/O aansluitingen	62	SD	
SPI poorten	1		

Checklist

Connectie met pc > 0,5 MBit/s	√
> 4 I/O pinnen	√
Extern geheugen	√

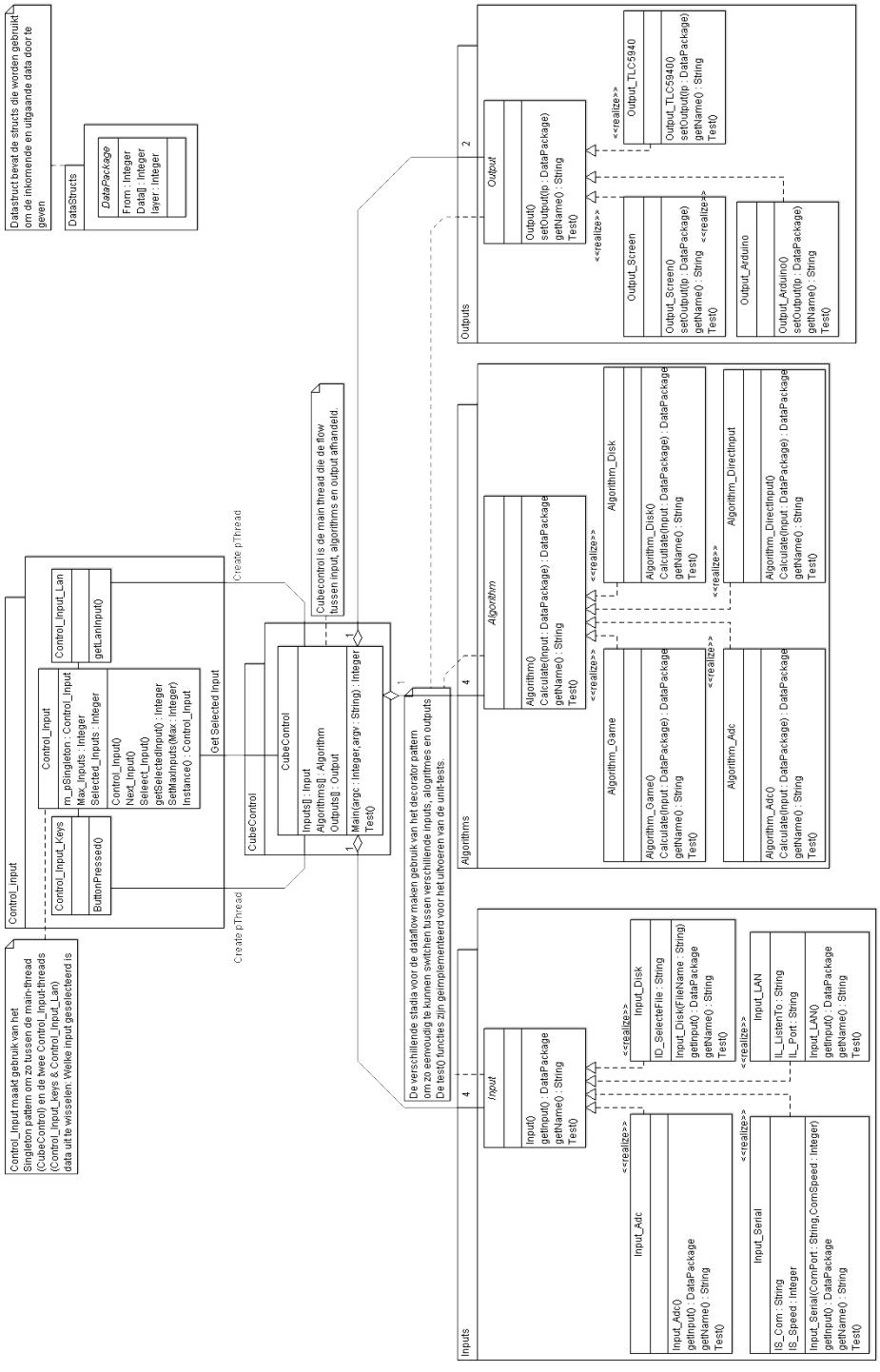
Kosten

Ontwikkelbord	€	169,00
Ontwikkeltools	€	0,00
Totaal	€	169,00

B.5.3 Conclusie

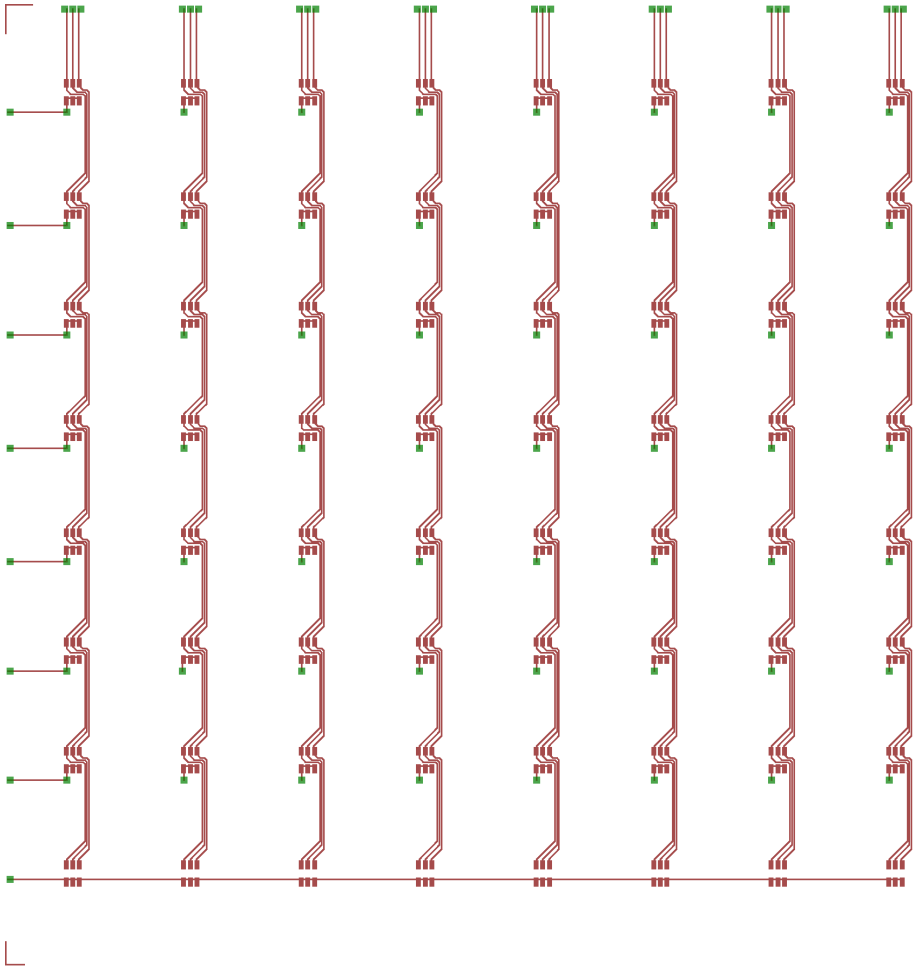
De keuze voor dit bord is omdat ik tijdens het onderzoek erachter kwam dat deze ontwikkelborden aanwezig waren bij TASS en incl. ontwikkelomgeving. Op het bord kan gebruik worden gemaakt van embedded linux, hierdoor is het eenvoudig om voor het apparaat software te ontwikkelen.
Eén van de nadelen is dat de grote van het geheel behoorlijk is, er moet nog onderzocht worden of dit past met de nog te bouwen prints in de kubus. Ook geeft het gebruik van embedded Linux iets meer overhead dan de dedicated embedded software.

C UML-Diagram CubeControl



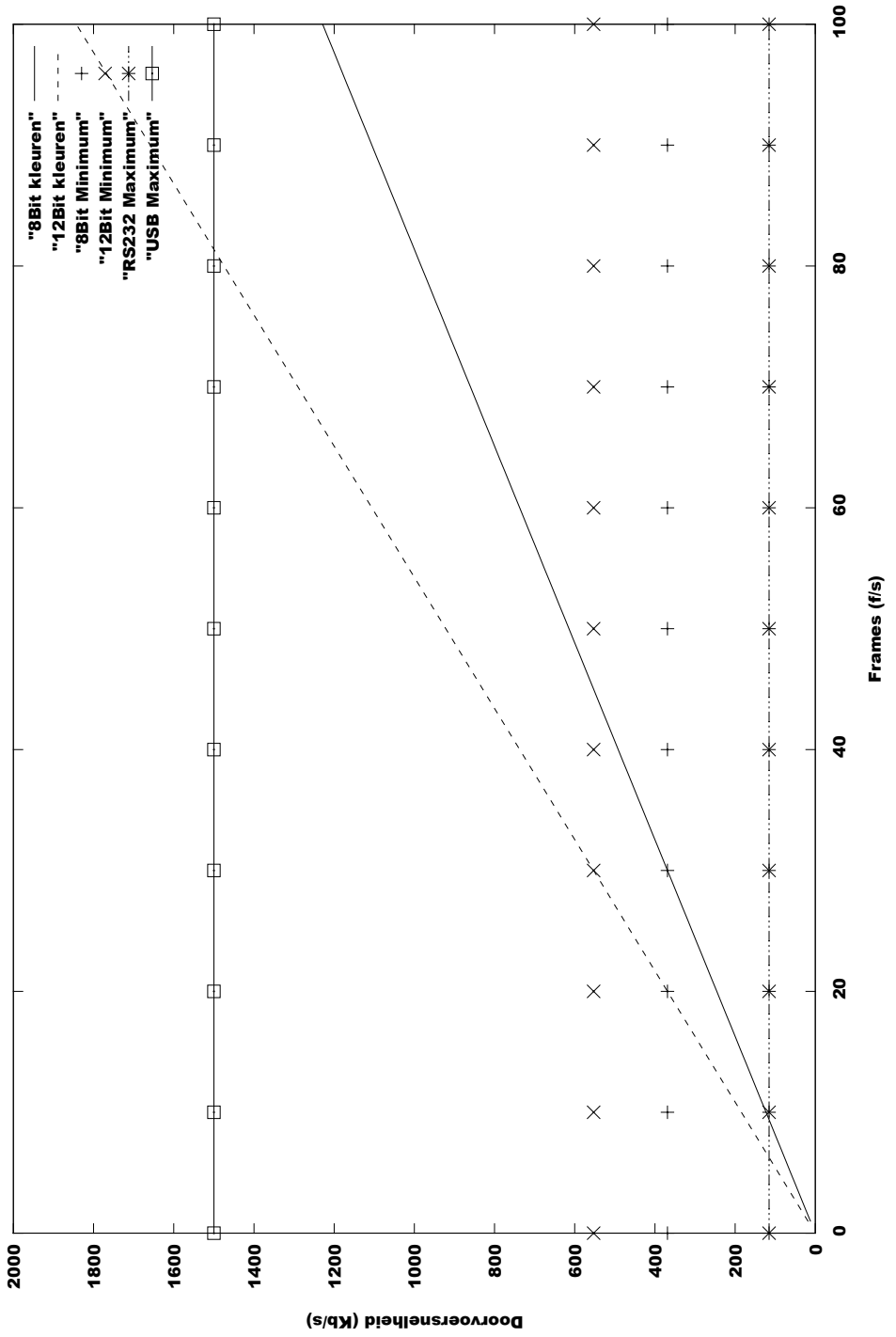
Figuur C.1: UML-diagram CubeControl

D Ontwerp Glazen Printplaat



Figuur D.1: Glazen Printplaat Kubus

E Onderzoek snelheden



Figuur E.1: ledaantal × bits × fps