

# Ontwikkelen Linux driver voor Blackmagic video capturekaart

## Afstudeerverslag

<i>Afstudeerder:</i>	Maarten Hamburg
<i>Studentnummer:</i>	20064311
<i>Bedrijf:</i>	CS Engineering
<i>Examinatoren:</i>	J.J Visser J.P.M de Vreught
<i>Opleiding:</i>	Technische informatie aan de Haagse hogeschool
<i>Locatie en datum:</i>	Waddinxveen, 4-6-2014

## Referaat

M.W.M Hamburg, Ontwikkelen Linux driver voor Blackmagic video capturekaart, afstudeerverslag  
Technische informatie aan de Haagse hogeschool.

Tijdens de uitvoering van dit project is door middel van reverse engineering inzicht gekregen in de werking van de Blackmagic DeckLink Studio capturekaart om de huidige driver compatible te maken met het Video4Linux2 framework.

### *Descriptoren*

- Drivers
- Linux besturingssysteem
- Capturekaart
- Afstudeerverslagen
- Technische Informatica

## Voorwoord

Dit document is het afstudeerverslag voor het project “Ontwikkelen Linux driver voor de Blackmagic video capturekaart”. Van de lezer van dit document wordt verwacht enige achtergrond kennis te hebben op het gebied van Linux, Drivers en de interactie met de hardware.

Ik heb met veel plezier en enthousiasme aan dit project gewerkt.

Graag wil ik de volgende mensen bedanken die hebben bijgedragen aan het uitvoeren van dit project.

- **Andre Canrinus:** voor de begeleiding tijdens dit project. Hij heeft mij indien dit noodzakelijk was de juiste richting op gestuurd.
- **Daan van Heteren:** voor het beschikbaar stellen van tijd om dit project uit te voeren.
- **John Visser:** voor de feedback die ik tijdens dit project heb gekregen.
- **Hans de Vreught:** voor de feedback op dit document.

Maarten Hamburg

Waddinxveen, 4-6-2014

## Samenvatting

DCT Net levert apparatuur en de infrastructuur aan kerken voor het online uitzenden van kerkdiensten. Omdat de vraag naar HD video kwaliteit toeneemt, is er door CS Engineering een nieuwe kerkzender ontwikkeld die deze kwaliteit mogelijk maakt. Deze kerkzender bevat een Blackmagic DeckLink Studio capturekaart waarvan de driver niet compatible is met het Video4Linux2 framework. Hierdoor wordt momenteel een extra conversie uitgevoerd die niet wenselijk is en vermindering van performance tot gevolg heeft. Allereerst is een projectplan opgesteld waarin een risico analyse is gemaakt. Deze risico analyse geeft de opdrachtgever inzicht in de risico's die mogelijk worden geacht tijdens de uitvoering van dit project.

Er is reverse engineering uitgevoerd om informatie te verkrijgen over de Blackmagic driver en de Blackmagic DeckLink Studio capturekaart. Deze informatie is noodzakelijk om de nieuwe driver te kunnen ontwikkelen en dus compatibiliteit met het Video4Linux2 framework te verkrijgen.

Tevens zijn er ontwerpen voor de nieuwe driver gemaakt. Met deze ontwerpen kan de nieuwe driver worden ontwikkeld.

Alhoewel de driver nog in ontwikkeling, is er aangetoond dat het haalbaar is deze driver volledig compatible te krijgen met het Video4Linux2 framework.

## Inhoudsopgave

<b>VOORWOORD .....</b>	<b>2</b>
<b>SAMENVATTING.....</b>	<b>3</b>
<b>1. INLEIDING .....</b>	<b>5</b>
<b>2. ORGANISATIE.....</b>	<b>6</b>
<b>3. OPDRACHTOMSCHRIJVING .....</b>	<b>8</b>
3.1. AANLEIDING .....	8
3.2. PROBLEEMSTELLING .....	8
3.3. DOELSTELLING .....	9
<b>4. INCEPTIEFASE.....</b>	<b>10</b>
<b>5. ELABORATIEFASE .....</b>	<b>14</b>
5.1. REVERSE ENGINEERING .....	14
5.2. PROTOTYPE .....	24
5.3. HUIDIGE ARCHITECTUUR .....	25
5.4. NIEUWE ARCHITECTUUR.....	28
5.5. ONTWIKKELRAPPORT.....	30
<b>6. CONSTRUCTIEFASE.....</b>	<b>35</b>
6.1. TESTEN.....	37
<b>7. EVALUATIE.....</b>	<b>39</b>
7.1. PRODUCT EVALUATIE .....	39
7.2. PROCES EVALUATIE .....	42
<b>BIBLIOGRAFIE.....</b>	<b>43</b>
BIJLAGE 1:PROJECTGEGEVENS .....	48
BIJLAGE 2: AFSTUDEERPLAN .....	49

## 1. Inleiding

DCT Net is een dochteronderneming van Communications Security Net B.V. (CS Net). Vanuit DCT Net worden producten geleverd aan kerken voor het streamen van video en audio uitzendingen, met als doel geïnteresseerden die niet in staat zijn naar de kerk te gaan toch betrokken te houden bij de kerkdiensten.

In dit document zal wanneer gesproken wordt over een kerkzender de aanduiding zender worden gebruikt. Een zender is een computer die is ingericht op gebruikersgemak. Hierdoor hoeft de zender enkel aangezet te worden om automatisch beeld en/of geluid op te nemen en te streamen naar de servers van DCT Net.

Bij de kerken die reeds beeld uitzenden neemt de vraag naar HD beeldkwaliteit toe. Om te voldoen aan deze wens is er door DCT Net een nieuwe zender ontwikkeld waarmee deze hogere beeldkwaliteit kan worden bereikt.

Vanuit DCT Net is gekozen voor deze Blackmagic DeckLink Studio capturekaart vanwege de beeldkwaliteit alsmede prijs van de capturekaart. Daarnaast was het bij de keuze van de capturekaart doorslaggevend dat deze kaart een SDI ingang heeft.

De Blackmagic DeckLink Studio capturekaart die in deze zender zit, wordt echter niet door het Video4Linux2 framework ondersteund. Hierdoor wordt momenteel op applicatie niveau een extra conversie uitgevoerd. De huidige conversie levert een vermindering in de performance van de zender op. Hierdoor wordt de ruimte voor hogere beeldkwaliteit beperkt.

Om dit probleem te verhelpen zal er een driver voor de Blackmagic DeckLink Studio capturekaart worden gemaakt die compatibiliteit met het Video4Linux2 framework mogelijk maakt.

De opbouw van dit document is als volgt:

In hoofdstuk 2 wordt achtergrond informatie gegeven over de organisatie waarbinnen dit project wordt uitgevoerd.

Hoofdstuk 3 geeft de opdrachtomschrijving voor het ontwikkelen van de driver weer. Ook wordt hier de aanpak alsmede het resultaat weergegeven.

In hoofdstuk 4 worden de uitgevoerde activiteiten in de inceptiefase beschreven. Dit omvat de risico analyse en het opstellen van de kandidaat systeemeisen.

Hoofdstuk 5 geeft de uitgevoerde activiteiten in de elaboratiefase weer. De volgende activiteiten zijn in dit hoofdstuk te vinden:

- Beschrijving reverse engineering.
- Beschrijving van de huidige en nieuwe architectuur.
- De ontwerpen voor de nieuwe driver.

In hoofdstuk 6 wordt beschreven welke stappen worden doorlopen om de driver daadwerkelijk te ontwikkelen.

Hoofdstuk 7 geeft een evaluatie van het project. Dit hoofdstuk is verdeeld in twee paragrafen namelijk de evaluatie van de producten en de proces evaluatie.

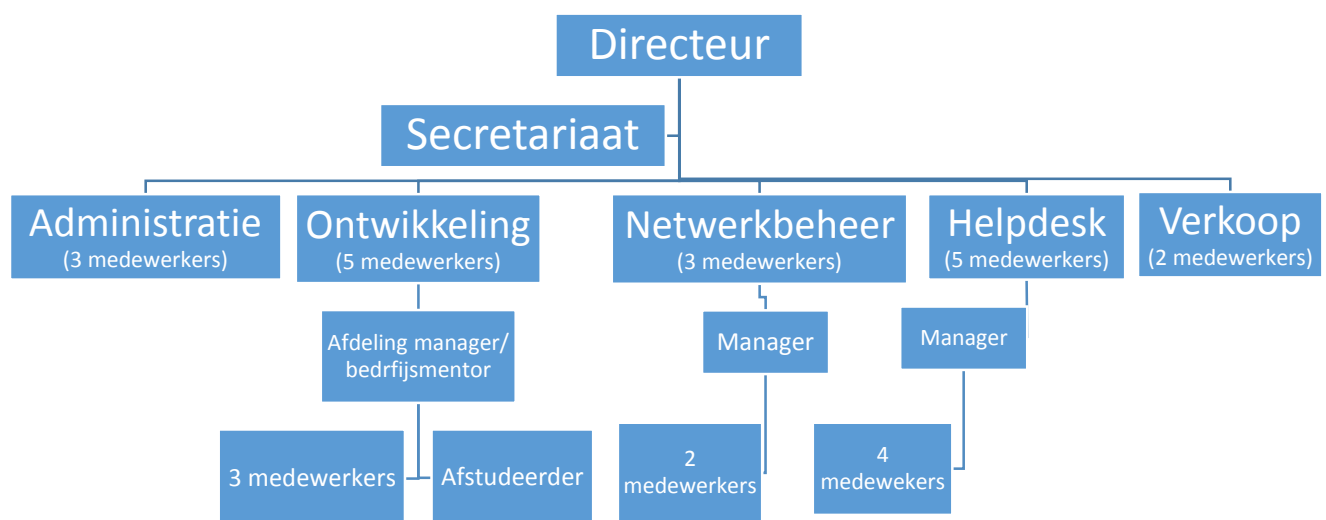
## 2. Organisatie

Communications Security Net B.V. (CS Net) is een IT bedrijf dat is gevestigd in Waddinxveen en gespecialiseerd is in het onderhouden van diverse infrastructuren en het ontwikkelen van maatwerk software.

Binnen CS Net worden de volgende kern activiteiten uitgevoerd:

- Onderhoud en aanleg van diverse technische netwerkinfrastructuren voor het midden- en klein bedrijf.
- Ontwikkeling van maatwerk software voor diverse branches waaronder communicatie software voor de hypothecaire sector.
- In-house helpdesk voor ondersteuning en installatie van software en netwerk diensten.

CS Net is een bedrijf dat behoort tot het midden en kleinbedrijf. Bij CS Net zijn momenteel 20 medewerkers in dienst. In Figuur 1 is een organigram weergegeven van de organisatie.



Figuur 1 Organigram organisatie CS Net

Zoals in Figuur 1 is te zien, ben ik werkzaam op de afdeling ontwikkeling. Deze afdeling valt onder de handelsnaam CS Engineering. Sinds maart 2013 ben ik werkzaam op deze afdeling. De manager van de afdeling, Andre Canrinus, is tevens mijn bedrijfsmentor.

Binnen de afdeling ontwikkeling worden maatwerk software pakketten gemaakt voor diverse branches waaronder diverse software ten behoeve van de kerkomroep.

Via het platform kerkomroep kunnen online kerkdiensten worden bekeken of beluisterd. Naast de website Kerkomroep [1] zijn er vanuit CS Engineering ook applicaties gemaakt voor Android [2] en Apple [3] smartphones en tablets waarmee de diensten kunnen worden bekeken of beluisterd.

DataCast Network (DCT Net) is een dochter onderneming van Communication Security Net B.V. en biedt services voor het streamen van gedigitaliseerde audio en videosignalen. DCT Net is de operationele partner voor stichting Intermediair Kerkomroep Nederland (sIKN). sIKN biedt oplossingen voor kerken om kerkdiensten live uit te kunnen zenden zodat deze via internet

beluisterd dan wel bekeken kunnen worden. Bij sIKN zijn momenteel meer dan 1000 kerken aangesloten.

DCT Net levert voor deze dienst de infrastructuur en de apparatuur om de kerkdienst uit te zenden en te ontvangen.

Zowel CS Engineering als CS Net zijn handelsnamen van Communications Security Net B.V. te Waddinxveen.



### 3. Opdrachtoomschrijving

#### 3.1. Aanleiding

Hieronder wordt de aanleiding weergegeven zoals deze in Bijlage 2: Afstudeerplan is opgenomen.

DCT Net heeft voor het bekijken van video diensten een nieuwe zender aangeschaft. Deze zender maakt gebruik van een Blackmagic HD capturekaart. Voor het streamen van video wordt binnen de organisatie FFMPEG gebruikt. FFMPEG is een cross-platform applicatie voor het opnemen, converteren en streamen van audio en video. Daarnaast wordt video gebruikt in combinatie met Video4Linux2. Video4Linux2 is een video capture API voor Linux. Hiermee kan video van diverse recorders worden gecaptured.

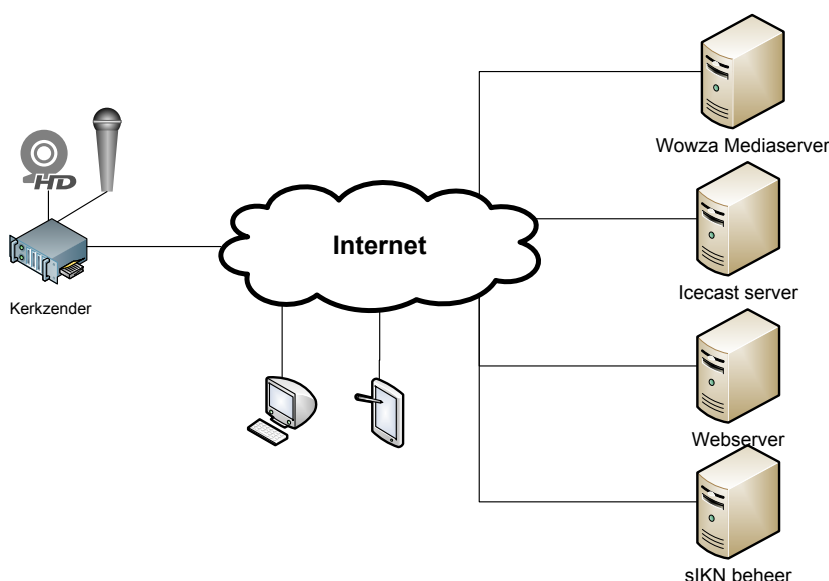
#### 3.2. Probleemstelling

Hieronder wordt de probleemstelling weergegeven zoals deze in Bijlage 2: Afstudeerplan is opgenomen.

De huidige driver van Blackmagic geeft als video data, onbewerkte video data, echter kan FFmpeg niet met deze onbewerkte video data overweg. Hiervoor is momenteel een eigen implementatie ontwikkeld waardoor de raw video data wordt omgezet naar video4linux2 ten behoeve van FFmpeg, echter wordt met deze implementatie de video data 2x meer verwerkt dan wenselijk is. Dit zorgt voor een ongewenste hogere processor load en geheugen gebruik.

Bovenstaande probleemstelling is niet goed geformuleerd, dit komt doordat bij de aanvang van het project onvoldoende informatie was verzameld over het daadwerkelijke probleem. De essentie van het probleem is echter wel zoals hierboven is geformuleerd. Het probleem is namelijk dat de Blackmagic driver niet compatible is met het Video4Linux2 framework. Hierdoor is het niet mogelijk direct de videodata door te geven naar FFmpeg. Hiervoor is een applicatie gemaakt die ervoor zorgt dat de video data wel kan worden verwerkt door FFmpeg. Door deze extra conversie wordt de videodata 2x meer verwerkt dan wenselijk is. Deze extra conversie zorgt voor een ongewenste hogere processor load en geheugen verbruik.

In Figuur 2 is een globaal overzicht weergegeven van de huidige architectuur.



Figuur 2 Overzicht van de huidige infrastructuur

In Figuur 2 is te zien dat op de zender een videobron en een audiobron zijn aangesloten. Deze zender is voorzien van de Blackmagic DeckLink Studio capturekaart. De zender streamt zowel video en audio via het internet naar de servers van DCT Net waarna de stream via internet kan worden geopend op diverse apparaten.

### 3.3. Doelstelling

Hieronder wordt de doelstelling weergegeven zoals deze in Bijlage 2: Afstudeerplan is opgenomen.

Het doel van de opdracht is om met een driver voor de Blackmagic capturekaart de onbewerkte video data direct om te zetten naar video4linux2 met verminderd gebruik van de resources.

#### Aanpak

Als project aanpak is gekozen de methodiek RUP te gebruiken. Hier is voor gekozen omdat RUP een methodiek is die toegespitst is op het ontwikkelen van software. Daarnaast geeft deze methodiek duidelijke handvatten, structuur en richtlijnen om een project als deze uit te voeren.

#### Resultaat

Tijdens de uitvoering van dit project is reverse engineering uitgevoerd op de Blackmagic driver. Door deze reverse engineering is inzichtelijk geworden hoe de aansturing van deze driver is. Ook is gedocumenteerd hoe de aansturing voor deze capturekaart is en welke I/O controls worden gebruikt. Er is tevens een prototype gemaakt die de compatibiliteit met het Video4Linux2 framework aantoonst. De driver is momenteel nog in ontwikkeling, maar er is inmiddels wel aangetoond dat het haalbaar is de driver volledig Video4Linux2 compatible te maken.

## 4. Inceptiefase

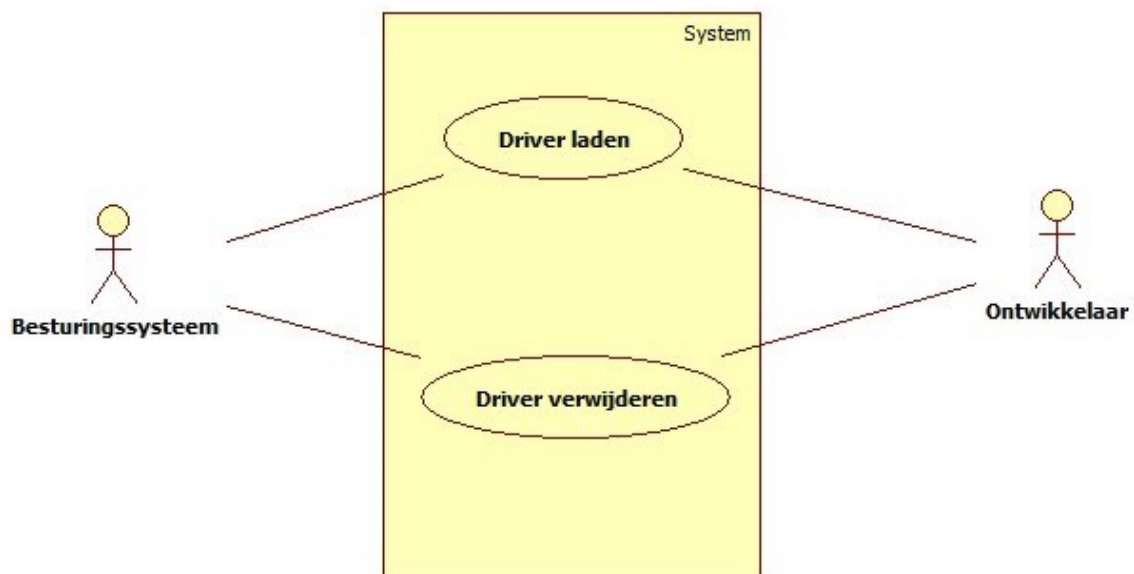
In deze fase is het vision document (plan van aanpak) gemaakt en zijn er functionele kandidaat systeemeisen in kaart gebracht. Het betreft in deze fasen alleen het opstellen van functionele kandidaat systeemeisen. Aan de hand van deze kandidaat systeemeisen kan later een use-case diagram worden opgesteld. De volledige functionele systeemeisen zullen in de elaboratiefase worden uitgewerkt nadat de reverse engineering is uitgevoerd.

In Tabel 1 zijn de eerste functionele kandidaat systeemeisen weergegeven.

Functionele kandidaat systeemeisen	
<b>Driver laden</b>	Via modprobe of insmod wordt de driver geladen in de kernel.
<b>Driver verwijderen</b>	De driver wordt via rmmod verwijderd uit de kernel

Tabel 1 Functionele kandidaat systeemeisen

Ook is in deze fase een use-case diagram gemaakt van de belangrijkste use-cases. Het gaat hier in dit geval om het laden van de driver en het verwijderen van de driver zoals weergegeven in Figuur 3.



Figuur 3 Eerste use-case-diagram

In Figuur 3 is de eerste versie van het use-case diagram weergegeven. Omdat een use-case diagram op zich weinig informatie verschaft wordt dit aangevuld met een use-case beschrijving die is weergegeven in Tabel 2 en Tabel 3.

Use-case beschrijving	
Naam	Driver laden
Samenvatting	De driver zal door het besturingssysteem of ontwikkelaar in de kernel worden geladen
Actoren	Besturingssysteem, Ontwikkelaar
Aannamen	<ul style="list-style-type: none"> <li>De zender is ingeschakeld</li> <li>De actor beschikt over root rechten</li> <li>De actor ontwikkelaar beschikt over een terminal</li> </ul>
Beschrijving	<ol style="list-style-type: none"> <li>De actor geeft het commando aan modprobe om de driver te laden</li> <li>Modprobe controleert of de driver geladen kan worden. Indien dit lukt, retourneert modprobe een 0. Zo niet treedt een uitzondering [ Fout bij laden ] op.</li> </ol>
Uitzonderingen	<ul style="list-style-type: none"> <li>[ fout bij laden ] De driver kan niet worden geladen omdat deze reeds in de kernel is geladen of een andere fout is opgetreden. De waarde 1 wordt geretourneerd.</li> </ul>
Resultaat	De driver is geladen in de kernel.

Tabel 2 Use-case beschrijving driver laden

In Tabel 2 is een beschrijving gegeven voor de use-case “driver laden”. De naam zoals in de bovenstaande tabel is weergegeven komt overeen met de naam van de use-case zoals ook in het use-case diagram is weergegeven. De samenvatting bevat de beschrijving van de uit te voeren use-case. De aannames beschrijven de toestand van het systeem bij de start van de use-case. In dit specifieke geval zijn er 3 aannames die voor aanvang van deze use-case voltooid moeten zijn. Bij de beschrijving wordt een volledige beschrijving gegeven van de interactie tussen het systeem en de actoren. Onder uitzonderingen worden condities verstaan die optreden tijdens het uitvoeren van de use-case. Het resultaat beschrijft het systeem na uitvoering van de use-case. In dit geval is de driver dan geladen [4].

In Tabel 3 is de use-case beschrijving gegeven voor de use-case driver verwijderen.

Use case beschrijving	
Naam	Driver verwijderen
Samenvatting	De driver zal door het besturingssysteem of ontwikkelaar verwijderd worden uit de kernel.
Actoren	Besturingssysteem, ontwikkelaar
Aannamen	<ul style="list-style-type: none"> <li>De zender is ingeschakeld</li> <li>De driver is geladen</li> <li>De actor beschikt over root rechten</li> <li>De actor ontwikkelaar beschikt over een terminal</li> </ul>
Beschrijving	<ol style="list-style-type: none"> <li>De actor geeft het commando aan rmmod om de driver te verwijderen.</li> <li>Rmmod controleert of de driver verwijderd kan worden. Indien dit is geslaagd retourneert het rmmod een 0. Wanneer dit niet lukt, treedt uitzondering [ fout bij verwijderen ] op.</li> </ol>
Uitzonderingen	<ul style="list-style-type: none"> <li>[ fout bij verwijderen ] De driver kan niet worden verwijderd uit de kernel. Rmmod retourneert de waarde 1.</li> </ul>
Resultaat	De driver is uit de kernel verwijderd.

Tabel 3 use-case beschrijving driver verwijderen

Om de problemen die kunnen ontstaan tijdens de uitvoering van het project in kaart te brengen is er een risico analyse gemaakt. In deze analyse is gekeken naar externe en interne factoren die het slagen van het project in gevaar kunnen brengen. Het maken van een risico analyse is tevens één van de competenties die aangetoond moeten worden. Het gaat om de competentie “Praktische aspecten hanteren in (internationale) projecten”. De volledige lijst met risico’s is opgenomen in het vision document [5].

De waarde van de impact is te zien in Tabel 4. In Tabel 5 is de waarde van het risico weergegeven. Een risico wordt berekend door de formule  $risico = kans * impact$ . In Tabel 6 is de score voor het risico weergegeven.

Impact score	Impact
5	Rampzalig
4	Groot
3	Matig
2	Klein
1	Te verwaarlozen

Tabel 4 Impact score

Kans score	Kans
5	Zeer hoge kans
4	Hoge kans
3	Matig kans
2	Lage kans
1	Zeer lage kans

Tabel 5 Kans score

Risico score	Omschrijving
25	De score voor dit risico is zeer hoog.
15 - 20	De score voor dit risico is hoog.
6 - 12	De score voor dit risico is laag.
1 - 5	De score voor dit risico is zeer laag.

Tabel 6 Risico score

Hieronder zijn enkele risico’s die in het vision document [5] zijn opgenomen weergegeven. In Tabel 7 wordt het risico voor defecte hardware beschreven.

Defecte hardware	
Beschrijving	Door het gebruik of onvoorziene omstandigheden kan de hardware defect raken.
Kans	2
Impact	5
Risico	10
Kans verlaging	<ul style="list-style-type: none"> <li>- Wanneer de hardware wordt aangeraakt of gedemonteerd, richtlijnen voor elektrostatische ontlading in acht nemen door bijvoorbeeld een speciale polsband te gebruiken.</li> <li>- Geen vloeistoffen in de buurt van de zender houden.</li> </ul>
Mitigatie	<ul style="list-style-type: none"> <li>- Zorgen voor een reserve opstelling.</li> </ul>
Plan B	<ul style="list-style-type: none"> <li>- De reserve opstelling gebruiken.</li> </ul>

Tabel 7 Defecte hardware

In Tabel 7 wordt het risico voor het defect gaan van de hardware beschreven. De kans dat het risico optreedt, is in dit geval laag en ingeschat op 2.

De impact wanneer het risico optreedt, is ingeschat op 5. Dit komt omdat er mogelijk data verloren is gegaan en het daarnaast tijd kost om de reserve opstelling gereed te hebben. Hierdoor komt de risico score uit op 10.

Risico naam	Software raakt verloren
Beschrijving	Doordat het systeem dermate is ontregeld kan de gemaakte software niet meer worden benaderd. Hiermee wordt bedoeld dat bijvoorbeeld de partitie tabel/SATA schijf beschadigd is geraakt en daardoor de data niet kan worden benaderd.
Kans	2
Impact	5
Risico	10
Kans verlaging	<ul style="list-style-type: none"> <li>- Wanneer de zender wordt verplaatst zorgen dat de harde schijf niet meer actief is.</li> </ul>
Mitigatie	<ul style="list-style-type: none"> <li>- Regelmatig software opslaan op meerdere locaties buiten het systeem waardoor een historie kan worden aangemaakt.</li> </ul>
Plan B	<ul style="list-style-type: none"> <li>- Vorige versies van de software gebruiken.</li> </ul>

Tabel 8 Software raakt verloren

Het risico in Tabel 8 betreft het kwijtraken van reeds gemaakte software. Dit kan optreden wanneer bijvoorbeeld de partitie niet meer te benaderen is. Omdat de nieuwe driver direct op de zender wordt ontwikkeld, op een externe SATA disk kan de software verloren gaan. De kans dat dit gebeurt is klein, echter is de impact hoog. Dit komt doordat de software verloren is gegaan. Om de kans te verlagen dat dit risico optreedt, mag de zender alleen worden verplaatst wanneer deze volledig is uitgeschakeld. Hiermee kan bijvoorbeeld worden voorkomen dat de arm van de harde schijf schade maakt wanneer deze onverwacht wordt bewogen. De impact is rampzalig omdat hiermee de reeds gemaakte software weg is. Hierdoor kan het zijn dat het project opnieuw moet beginnen. Om de impact te verlagen moet de software regelmatig op meerdere locaties buiten het systeem worden opgeslagen. Wanneer het risico dan optreedt, kan de laatste versie van de software worden gebruikt.

## 5. Elaboratiefase

In deze fase wordt reverse engineering uitgevoerd om informatie te achterhalen die noodzakelijk is voor het maken van de nieuwe Video4Linux2 driver. Ook wordt in deze fase de huidige alsmede de nieuwe architectuur beschreven en worden de ontwerpen voor de nieuwe driver gemaakt. Het uitvoeren van de reverse engineering, en het beschrijven van de huidige is simultaan aan elkaar uitgevoerd. Dit komt omdat tijdens het uitvoeren van de reverse engineering ook de architectuur in kaart wordt gebracht.

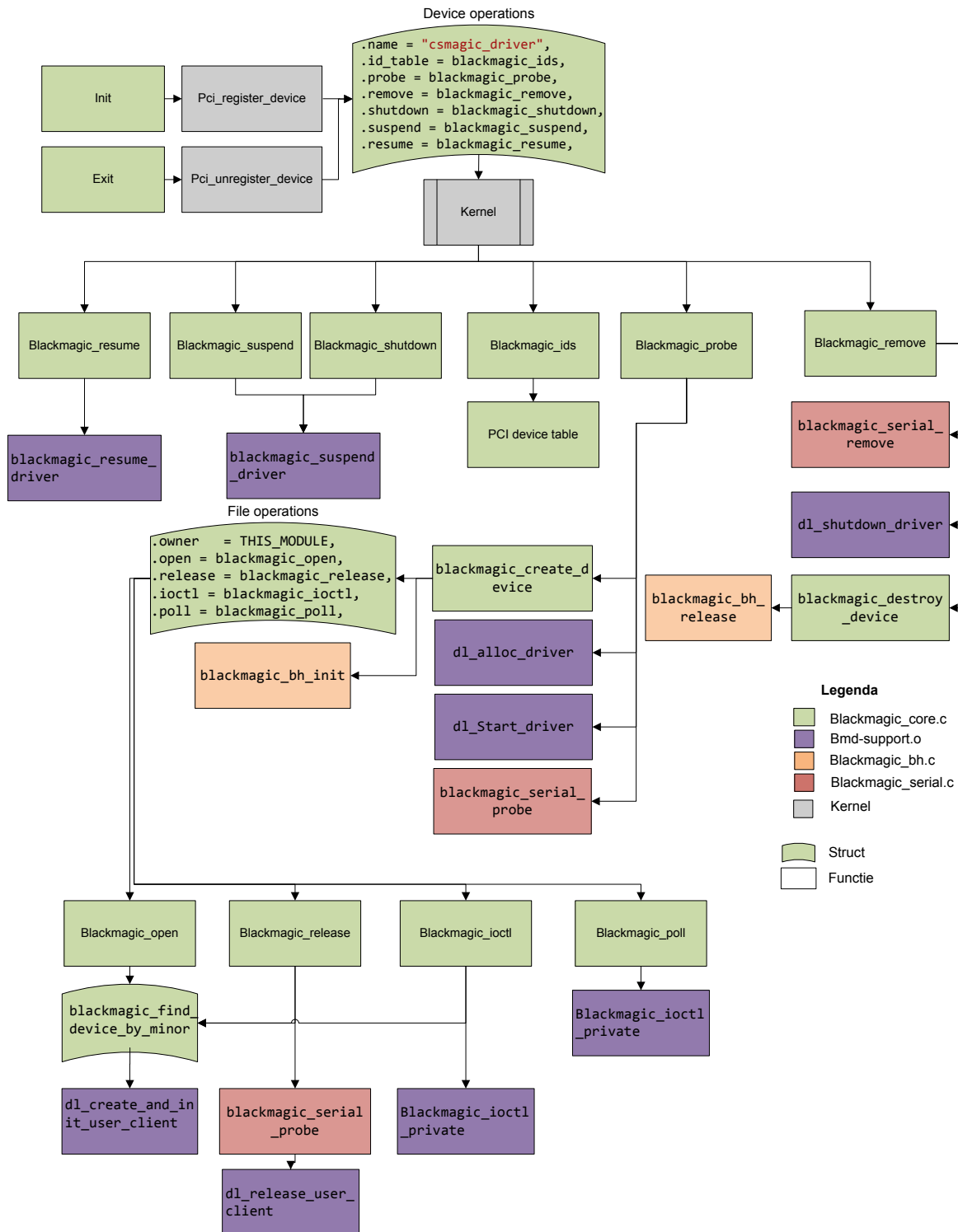
### 5.1. Reverse engineering

Door middel van reverse engineering moet voldoende informatie worden achterhaald om de nieuwe Video4Linux2 driver te kunnen ontwikkelen. Reverse engineering is: *“het onderzoeken van een product (meestal een stuk software of een communicatieprotocol) om daaruit af te leiden wat de eisen zijn waaraan het product probeert te voldoen, of om de precieze interne werking ervan te achterhalen”* [6]. De resultaten van de reverse engineering zijn vastgelegd in het analyserapport [7]. Door het uitvoeren en beschrijven van de reverse engineering wordt de competentie “achterhalen van het probleemdomein” aangetoond.

Van een deel van de Blackmagic driver is de source beschikbaar [8]. Hierdoor kan worden geanalyseerd wat de werking van de driver is en wat de samenhang is tussen diverse functies en structs binnen de driver. Deze verkregen informatie zal worden gebruikt voor het uitvoeren van de reverse engineering en de resultaten zullen worden gedocumenteerd in het analyserapport. De belangrijkste broncode zit in het bestand `blackmagic_core.c` omdat hierin de file operation `ioctl` voor de I/O controls is gedefinieerd.

Om een goed overzicht te krijgen van de structuur van de `blackmagic_core.c` is hiervan een overzicht gemaakt en dit is weergegeven in Figuur 4. In deze afbeelding is te zien welke functies en structs er in de `blackmagic_core.c` zitten. De enige uitzondering in deze figuur is de kernel. Deze is toegevoegd omdat de functies `blackmagic_resume`, `blackmagic_suspend`, `blackmagic_shutdown` en dergelijke worden aangeroepen door een interactie met de kernel. De device operations struct bevat enkel pointers naar de betreffende functie.

De device operations, file operations en `blackmagic_find_device_by_minor` zijn structs en zijn derhalve weergegeven als een ander type. De kleuren van de functie geven aan binnen welke source file de functie zich bevindt. Dit is tevens weergegeven in de legenda.



Figuur 4 Flow van de huidige Blackmagic driver

De file operations (fops) ioctl is een pointer naar de functie blackmagic\_ioctl. Deze functie wordt aangeroepen wanneer een I/O control wordt gegeven aan de driver. Echter is de functie blackmagic\_ioctl een wrapper om de functie blackmagic\_ioctl\_private die is gedefinieerd in het bmd-support.o object. Wat de implementatie is van deze en de andere functies binnen dit object bmd-support.o is, is niet bekend. Ook is momenteel niet bekend welke I/O controls er worden uitgevoerd en wat deze betekenen, hiervoor is de reverse engineering noodzakelijk.



Het is van belang te weten hoe de aansturing van de huidige Blackmagic driver werkt omdat er een conversie uitgevoerd moet worden tussen de Video4Linux I/O controls en de Blackmagic I/O controls. Onder andere door deze conversie wordt compatibiliteit met het Video4Linux2 framework mogelijk. Ook moet aan de driver de functionaliteit van het Video4Linux2 framework worden toegevoegd om de daadwerkelijk compatibiliteit te verkrijgen. Het gaat hier dan bijvoorbeeld om het aanmaken van een character device `/dev/video0`. Achtergrond informatie over het Video4Linux2 framework wordt later in dit hoofdstuk op pagina 23 gegeven.

Momenteel wordt in `/dev/` het character device `blackmagic0` aangemaakt met major nummer 10. Dit betekent dat het een miscellaneous character device is. Een miscellaneous device wordt gebruikt voor drivers die in te kleine aantallen worden uitgebracht om een eigen major nummer te krijgen [9].

Om te achterhalen welke I/O controls worden uitgevoerd door de driver is in de functie `blackmagic_ioctl` van de `blackmagic_core.c` een logregel toegevoegd. Door het toevoegen van de logregel worden de I/O controls die via `blackmagic_ioctl` naar de `blackmagic_ioctl_private` worden gestuurd in de systeemlog geschreven. Het loggen van de commando's gebeurt door de functie `dl_printk`.

De volledig gebruikte syntax die is gebruikt is: `dl_printk(DL_MSG "blackmagic ioctl cmd=0x%X arg=0x%X [%X] -> [%X]\n", cmd, arg, ervoor, erna )`. `DL_printk` is gedefinieerd in de source file `blackmagic_lib.c` en is een wrapper rond de kernel functie `vprintk`.

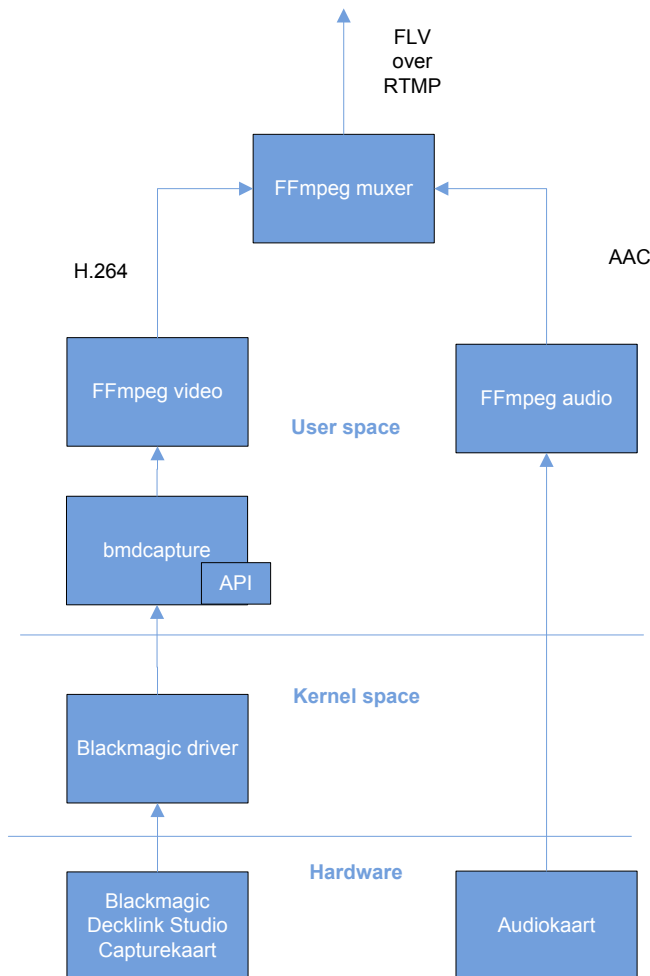
De waarde ervoor en erna is verkregen door een `copy_from_user` uit te voeren voor en nadat de `blackmagic_ioctl_private` is aangeroepen. Hiermee kan worden gecontroleerd of de waarde in de buffer `arg` is gewijzigd.

Om een I/O control naar de `blackmagic_ioctl_private` te sturen zijn vier parameters noodzakelijk:

- **Ddev:** bevat de device struct.
- **Filp:** Dit is struct file die een "open file" representeert. Deze pointer wordt aangemaakt door de kernel wanneer een open van het apparaat wordt uitgevoerd. Deze file pointer bevat onder andere de file\_operaties en de private data voor de driver [10].
- **Cmd:** is het commando dat wordt gebruikt om de hardware aan te sturen. Dit commando kan dus variëren afhankelijk van de uit te voeren actie.
- **Arg:** wordt doorgegeven in de vorm van een unsigned long ongeacht of er een integer of een pointer wordt meegegeven [11].

Vanuit Blackmagic is het programma `bmdcapture` geleverd dat door CS Engineering is aangepast om de video data door te kunnen geven aan FFmpeg. Van dit programma `bmdcapture` is de source beschikbaar en dit programma wordt momenteel bij één klant gebruikt.

`Bmdcapture` maakt gebruik van de Blackmagic SDK (Software Development Kit) en API (Application Programming Interface). De SDK en API kunnen worden gedownload van de website van Blackmagic [12]. In Figuur 5 is te zien hoe momenteel de conversie verloopt van de driver naar FFmpeg.



Figuur 5 Huidige werking van de driver

De capturekaart geeft de video data door naar de Blackmagic driver. In werkelijkheid plaatst de capturekaart de data in het geheugen en geeft door aan de driver waar de data is geplaatst. Vervolgens haalt bmdcapture via de API de video data uit het geheugen waarna een Video4Linux2 header voor de video data wordt geplaatst. Het doorgeven van de video data aan FFmpeg gebeurt via een pipe. Een pipe redirect data zodat de data kan worden benaderd door een andere applicatie [13]. FFmpeg zal vervolgens encoding naar H.264 uitvoeren om het daarna verder te verwerken.

Wanneer het programma bmdcapture in capture mode wordt gezet levert dit dermate veel I/O controls op in de log, dat de functie/doel van een I/O control niet eenvoudig is te achterhalen. Deze logregels zijn in de systeemlog geplaatst door de toevoeging van de logregel in de blackmagic\_ioctl.

Om uiteindelijk de I/O controls te kunnen linken aan de Blackmagic API functies, is er een programma genaamd Capture gemaakt dat kan worden gestart met parameters. Dit programma is gebaseerd op de source van het programma bmdcapture. Door het programma Capture te parameteriseren is het mogelijk om dit programma vanaf de console te starten met specifieke parameters waardoor een specifieke functie van de API wordt uitgevoerd. Door tevens in de betreffende functies van dit programma Capture log regels toe te voegen kan worden herleid welke regel in de systeemlog bij een API functie hoort. Zodoende kan een I/O control worden gelinkt aan een API functie.

Als het programma Capture wordt uitgevoerd zonder parameters wordt enkel de initialisatie van de capturekaart uitgevoerd.

De initialisatie van de capturekaart gebeurt via de API functie *CreateDeckLinkIteratorInstance()*. De logregels die te zien zijn in Voorbeeld 1 zijn verkregen door het initialiseren van de capturekaart en de toegevoegde logregels in de functie *blackmagic\_ioctl*.

```
cmd=0x10000 arg=0xBF9DD49C [0] -> [BB]
cmd=0x99A arg=0xBF9DD424 [10000] -> [0]
cmd=0x10000 arg=0xBF9DD554 [BF9DD564] -> [BB]
cmd=0x990 arg=0x9FF23A4 [0] -> [7EA000]
cmd=0x940 arg=0xBF9DD5FC [B7478710] -> [A120]
cmd=0x10000 arg=0xBF9DD654 [9FF2390] -> [BB]
cmd=0x990 arg=0x9FF23A4 [0] -> [7EA000]
cmd=0x994 arg=0xBF9DD734 [0] -> [1]
cmd=0x990 arg=0x9FF2530 [72657620] -> [7EA000]
cmd=0x990 arg=0xBF9DD33C [B75E1FF4] -> [7EA000]
cmd=0x990 arg=0xBF9DD33C [B751A9EF] -> [7EA000]
```

Voorbeeld 1 Losse functie voor debug doeleinden

In de log van Voorbeeld 1 is te zien dat er diverse regels in de systeem log zijn geschreven. Het gaat hier om het commando, argument en de waarde ervoor en erna.

Zoals eerder is aangegeven is het programma Capture te starten met parameters. De parameter die wordt meegegeven, bepaalt welke API functie er wordt aangeroepen en dus welke I/O controls worden uitgevoerd. Sommige parameters hebben een extra parameter nodig zoals een displaymode. Een deel van de beschikbare parameters is weergegeven in Tabel 9. De volledige lijst is te vinden in het analyserapport [7].

Capture parameter	API functie	Parameters
	<i>CreateDeckLinkIteratorInstance()</i>	
<b>GetModelName</b>	<i>deckLink-&gt;GetModelName()</i>	deckLinkModelName
<b>setbmdVideoConnectionSDI</b>	<i>deckLinkConfig-&gt;SetInt()</i>	bmdDeckLinkConfigVideoInputConnection, value
<b>getbmdVideoConnection</b>	<i>deckLinkConfig-&gt;GetInt()</i>	""
<b>setbmdVideoConnectionComposite</b>	<i>deckLinkConfig-&gt;SetInt()</i>	""
<b>doesSupportVideoMode</b>	<i>deckLinkInput-&gt;DoesSupportVideoMode()</i>	selectedDisplayMode, pixelFormat, bmdVideoInputFlagDefault, &result, NULL

Tabel 9 Parameters voor het programma capture

Om de displaymodes te verkrijgen is in het programma Capture een functie *displayModeGetName()* gemaakt. Deze functie itereert door alle displaymodes die door de capturekaart worden ondersteund en geeft de waarden van de displaymodes zoals deze in Tabel 10 zijn opgenomen. Deze waarde is bijvoorbeeld 1215312441.

Wanneer bijvoorbeeld de parameters *doesSupportVideoMode* en 1215312441 worden meegegeven aan het programma Capture zal de capturekaart via de API functie *deckLinkConfig->DoesSupportVideoMode()* controleren of videomode wordt ondersteund door de capturekaart. Het uitvoeren van deze API functie resulteert in informatie in de systeemlog zoals is weergegeven in Voorbeeld 2.

```
cmd=0x10000 arg=0xBF80511C [0] -> [BB]
cmd=0x99A arg=0xBF8050A4 [10000] -> [0]
cmd=0x10000 arg=0xBF8051D4 [BF8051E4] -> [BB]
cmd=0x990 arg=0x937A3A4 [0] -> [7EA000]
cmd=0x940 arg=0xBF80527C [B74B3710] -> [A120]
cmd=0x10000 arg=0xBF8052D4 [937A390] -> [BB]
cmd=0x990 arg=0x937A3A4 [0] -> [7EA000]
cmd=0x994 arg=0xBF8053B4 [0] -> [1]
cmd=0x990 arg=0x937C5B0 [0] -> [7EA000]
cmd=0x990 arg=0xBF804FBC [B761CFF4] -> [7EA000]
cmd=0x990 arg=0xBF804FBC [B75559EF] -> [7EA000]
cmd=0xA06 arg=0xBF805334 [9] -> [9]
```

Voorbeeld 2 Systeemlog van de initialisatie en aanroep doessupportdisplaymode

Zoals te zien in het groene deel van Voorbeeld 2, is nu naast de initialisatie van de capturekaart ook een commando 0xA06 in de log geschreven. Dit commando behoort bij de API functie *deckLinkConfig->DoesSupportVideoMode ()*. De waarde 9 staat in dit geval voor de waarde van de displaymode. De diverse displaymodes zijn te vinden in het analyserapport [7] en een deel van de displaymodes is te zien in Tabel 10.

Waarde	Displaymode	Hexadecimaal	Tekst	Mode	Frames
5	842231923	32337073	23ps	1080 p	23.98
7	842297459	32347073	24ps	1080 p	24
8	1215312437	48703235	Hp25	1080 p	25
9	1215312441	48703239	Hp29	1080 p	29.97
B	1214854448	48693530	Hi50	1080 i	50

Tabel 10 Displaymodes

De waarden van de displaymodes worden gebruikt om de juiste framerate en direct ook de juiste resolutie op de capturekaart in te stellen. Ook is achterhaald dat de displaymode het aantal frames per seconde bevat, maar ook of het progressive of interlaced scanning betreft. In de kolom Tekst is de framerate te zien. De waarde p of i in deze kolom geeft aan of het interlaced of progressive scanning betreft. De informatie is verkregen door de displaymode om te zetten naar een hexadecimale waarde en deze waarden op te zoeken in een ASCII tabel [14].

Omdat niet elke I/O control door deze aanpak kan worden gelinkt aan een API functie, bijvoorbeeld omdat er meerdere I/O controls worden uitgevoerd, is het object bmd-support.o gedisassembleerd. Dit is gedaan met het programma objdump [15]. Objdump is een disassembler voor Linux. Dit betekent dat gecompileerde data wordt omgezet naar assembler code [16]. Door “*objdump -D bmd-support.o*” uit te voeren wordt dit object gedisassembleerd. Dit betekent dat de assemblercode zichtbaar wordt gemaakt evenals instructies die worden uitgevoerd en de adressen waarop een instructie zich bevindt.

Wanneer de functie *blackmagic\_ioctl\_private(ddev->driver, filp->private\_data, cmd,arg)* wordt aangeroepen met de gegeven parameters vanuit de *blackmagic\_ioctl* dan worden de functieparameters via registers doorgegeven. De functieparameters voor de functie *blackmagic\_ioctl\_private* zijn in Tabel 11 weergegeven. Het betreft hier 32 bits registers.

Parameter	Register
Arg	%ecx
Cmd	%edx
Filp->private_data	%esi
Ddev->driver	%edi

Tabel 11 Assembly registers [17]

Vanuit de systeemlog die eerder is verkregen kan in de assemblercode worden achterhaald welke instructies hier worden uitgevoerd voor een I/O control. Hieronder wordt het achterhalen van de I/O control 0x910 beschreven.

- Het ioctl commando is de 3<sup>e</sup> parameter van de aanroep bmd\_ioctl\_private. Dit betekent dat de numerieke waarde op het moment van de aanroep te vinden is in register %edx.
- In de gedisassembleerde code van de bmd\_support.o kan nu worden gezocht naar instructies waarvan de inhoud van het register %edx wordt vergeleken.  
8dc: 81 fa 10 09 00 00 cmp \$0x910,%edx  
8e2: 0f 84 c5 0b 00 00 je 14ad <blackmagic\_ioctl\_private+0x14ad>
- blackmagic\_ioctl\_private+0x14ad geeft aan dat de functie die de implementatie van ioctl commando 0x910 bevat, begint op offset van 14ad van blackmagic\_ioctl\_private.  
14ad: 8b 3d 00 00 00 00 mov 0x0,%edi
- Door het aanroepen van dl\_access\_ok met de parameter bmd\_verify\_read wordt in dit geval gekeken of een blok geheugen van 4 bytes in user-space valide is [18].  
dl\_access\_ok is een wrapper voor de functie access\_ok en is gedefinieerd in blackmagic\_lib.c
- Diverse registers worden geladen om te gebruiken voor de aanroep van de functie videoInputOn  
14f0: e8 00 00 00 00 call 14f5 <blackmagic\_ioctl\_private+0x14f5>  
14f1: R\_X86\_64\_PC32\_ZN15UserClientClass12videoInputOnEj+0xfffffffffffffc
- Vervolgens worden 4 bytes met de call \_\_dl\_copy\_from\_user in kernel-space geplaatst.

Aan de ioctl 0x910 kan de mangled symbolische naam UserClientClass::videoInputOn worden toegekend. De naam kan worden herleid door de identifier uit Tabel 12 [19].

Identifier	Beschrijving
<b>_Z</b>	Alle mangled symbolen beginnen met de _Z
<b>N</b>	Het symbol is een nested symbol.
<b>15, 12</b>	Geven het aantal characters aan dat volgen
<b>E</b>	Dit is het einde van het symbol
<b>J</b>	Dit geeft aan dat de return value een unsigned int is.

Tabel 12 mangled name parameters

De via de objectdump verkregen functienamen en de verkregen commando's zijn vastgelegd in het analyserapport [7] waarvan een deel is weergegeven in Tabel 13.

Functienaam	Cmd	Value	Size	Type	Opmerking
<b>UserClientClass::videoInputOn</b>	0X910		4	Read	
<b>UserClientClass::audioInputstart</b>	0X936	Waarde van de audio channels, 2, 8, 16	12	Read	
<b>UserClientClass::audioInputStop</b>	0X937	0	-		
<b>UserClientClass::audioInputReadSamples::BlackmagicAudioInputReadParams::BlackmagicAudioInputReadResult</b>	0X938		40	Write	Komt alleen voor bij video en audio enabled

Tabel 13 I/O control mapping

In Tabel 13 is een deel te zien van de verkregen I/O controls. De functie namen zijn verkregen uit de assemblercode van het object bmd-support.o zoals hierboven is beschreven.

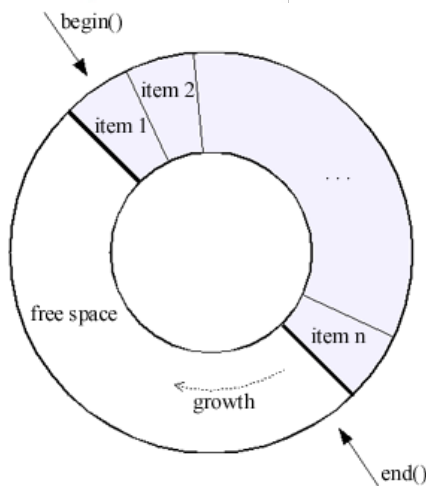
De kolom cmd is het bijbehorende commando zoals door Blackmagic wordt aangestuurd en is verkregen via de systeemlog.

De kolom size bevat het aantal bytes dat wordt gelezen of geschreven. De type is een read of een write. Dit is afgeleid door te kijken in de disassembly of bij dl\_access\_ok de parameter bmd\_verify\_write of bmd\_verify\_read is. bmd\_verify\_write is overeenkomstig met de kernel definitie VERIFY\_WRITE. Bmd\_verify\_read komt overeen met de kernel definitie VERIFY\_READ.

De verkregen Blackmagic I/O controls worden gebruikt om een mapping te kunnen maken tussen Video4Linux2 I/O controls en een Blackmagic I/O controls. Wanneer een I/O control bedoeld is voor Video4Linux2 zal worden bepaald of dit een instructie is die inderdaad naar het framework mag of naar de blackmagic\_ioctl moet gaan. Hierdoor kan de juiste aansturing van de capturekaart plaatsvinden.

Een essentieel onderdeel van de Blackmagic DeckLink Studio capturekaart is de manier waarop de videodata in het geheugen wordt geplaatst. Dit is van belang omdat de video data uiteindelijk door de FFmpeg uit het geheugen moet worden gehaald maar ook voor algemene achtergrond kennis.

FFmpeg of een andere applicatie allocceert bij het starten van een stream ruimte in het geheugen om de data kwijt te kunnen (pages). De Blackmagic driver maakt gebruik van DMA om de verkregen video data in het geheugen van de applicatie te plaatsen. In Figuur 6 is weergegeven hoe het buffer mechanisme werkt voor de Blackmagic capturekaart.



Figuur 6 Circulair geheugen [20]

Voor het gebruik van DMA door de Blackmagic driver wordt een pool van 30 geheugen buffers gemaakt. Wanneer er data naar het geheugen wordt geschreven zal dit in een van de aangemaakte DMA buffers zijn. Dit is zoals Figuur 6 aangeeft een ringbuffer. Wanneer er video data wordt gecaptured wordt dit naar een locatie in het geheugen geschreven. Wanneer vervolgens de data is verwerkt, wordt deze buffer geleegd.

Deze informatie is uiteindelijk verkregen door het analyseren van de log die is gemaakt na het toevoegen van logregels. Het gaat hier om de toegevoegde logregels in de `blackmagic_ioctl`. Door de Blackmagic driver wordt gebruik gemaakt van streaming DMA. Dit is te herkennen omdat in de `blackmagic_dma.c` de kernel functie `pci_map_single(pdev, address, size, direction)` wordt gebruikt [21]. De `direction` geeft de richting aan waarin de data geschreven moet worden. Deze waarden kunnen zijn:

- `PCI_DMA_TODEVICE`
- `PCI_DMA_FROMDEVICE`
- `PCI_DMA_BIDIRECTIONAL`
- `PCI_DMA_NONE`

Omdat de driver moet weten wanneer de data in het geheugen is geschreven door de capturekaart zijn er interrupt handlers beschikbaar. De afhandeling van interrupts bestaat uit 2 delen namelijk de “top half” en de “bottom half” interrupt handler. De top half interrupt handler werkt op een laag systeem niveau. Dit betekent bijvoorbeeld dat wanneer de capturekaart klaar is met schrijven van data naar het geheugen er een interrupt wordt gestuurd naar de top half interrupt handler. Op een later moment wordt deze interrupt verwerkt door de bottom half interrupt handler. De bottom half handler wordt geregistreerd door vanuit de `blackmagic_core.c` de functie `blackmagic_bh_init` aan te roepen. Hier wordt uiteindelijk een thread aangemaakt die kan worden geïnterrupteerd. Deze laatste bottom half interrupt handler wordt uitgevoerd door een scheduler. Dit betekent dat bijvoorbeeld in de eerstvolgende programma cyclus de interrupt wordt afgehandeld [22]. Deze interrupt handler is een shared interrupt handler. Dit houdt in dat het IRQ nummer (16) dat tijdens opstarten is verkregen gedeeld kan worden met een ander apparaat in dit geval is dat usb3. Dat het een shared interrupt handler is terug te vinden in de `blackmagic_core.c`. Hier wordt de functie `request_irq(pci_dev->irq, blackmagic_isr, SA_SHIRQ, dev->mdev.name, dev)` uitgevoerd. De parameter `SA_SHIRQ` geeft aan dat het om een shared interrupt gaat. In Tabel 14 is een overzicht gegeven van een aantal IRQ's die op de zender geregistreerd zijn waaronder IRQ 16.



IRQ	CPU								
	CPU0	CPU1	CPU2	CPU3	CPU4	CPU5	CPU6	CPU7	
0	42	0	0	0	0	1	0	0	IO-APIC-edge timer
1	1	0	0	0	1	0	0	0	IO-APIC-edge i8042
8	1	0	0	0	0	0	0	0	IO-APIC-edge rtc0
9	0	0	0	0	0	0	0	0	IO-APIC-fastestoi acpi
16	31919	62	577	42	944	39	51	9	IO-APIC-fastestoi ehci_hcd: usb3, blackmagic0

Tabel 14 /proc/interrupts

In Tabel 14 is te zien dat er een interrupt handler genaamd blackmagic0 geregistreerd voor IRQ 16 samen met usb3. De interrupts zijn voor de interrupt handler te herkennen doordat het IRQ nummer en device\_id wordt meegegeven bij registratie. Het meegeven van een device id is noodzakelijk voor een shared interrupt en hierdoor kan de juiste interrupt handler de IRQ afhandelen.

## Video4Linux2

Nu globale informatie is verzameld over de werking van de Blackmagic DeckLink Studio capturekaart en het Linux besturingssysteem, is het van belang inzicht te krijgen in het Video4Linux2 framework. Video4Linux2 is een device API en driver framework voor ondersteuning van media apparaten [23].

Om van dit framework gebruik te kunnen maken is het noodzakelijk dat er door de driver een character device wordt aangemaakt. Deze character devices kennen veelal de file operations:

- Open
- Close
- Read
- Write
- Ioctl

Het Video4Linux2 character device zal op de locatie /dev/ te vinden zijn als video0. Echter om compatible te zijn met het Video4Linux2 framework moet aan dit apparaat een major nummer 81 worden toegekend. Dit is een nummer dat automatisch wordt toegekend aan het apparaat als deze goed is geïnitieerd door het Video4Linux2 framework. Het minor nummer voor het apparaat kan variëren tussen de 0 en 255 [24].

Om de Blackmagic capturekaart te laten herkennen door het Video4Linux2 framework moeten er diverse functies worden toegevoegd aan de blackmagic\_core.c. Om te achterhalen hoe een Video4Linux2 driver werkt is er gekeken naar bestaande Video4Linux2 drivers. Er is bijvoorbeeld gekeken naar de source file vivi.c en de cx23885 driver. Deze drivers zijn te vinden in de linux-source [25]. Vivi.c is een virtuele video driver waarmee een video device kan worden geëmuleerd om te communiceren met het Video4Linux2 framework.

Voor het verkrijgen van de compatibiliteit met het Video4Linux2 framework moet de huidige Blackmagic fops worden vervangen door de Video4Linux2 fops.

De nieuwe fops zal een struct v4l2\_file\_operations zijn, terwijl dit voorheen een struct file\_operations was. Dit heeft tot gevolg dat de ingaande parameters van een functie gewijzigd dienen te worden. De functie blackmagic\_open bijvoorbeeld, heeft in de blackmagic\_core als parameters struct inode \*inode, struct file \*filp terwijl dit voor Video4Linux2 enkel struct file \*filp moet zijn.



Door zulke wijzigingen uit te voeren in de driver wordt deze compatible met het Video4Linux2 framework. Uiteraard dienen ook functies toegevoegd te worden om überhaupt het character device aan te maken. Hiervoor dient gebruik te worden gemaakt van functies als `V4l2_register_device`. Welke functies toegevoegd moeten worden zijn beschreven in de Video4Linux2 framework documentatie [26]. Voor het compatible maken zal voornamelijk worden gekeken naar hoe de cx23885 driver werkt. Deze driver wordt binnen CS Engineering ook gebruikt voor andere video toepassingen.

## 5.2. Prototype

In deze fase is ook een evolutionary [27] prototype gemaakt. Met dit prototype is aangetoond dat het mogelijk is om het device te benaderen via het Video4Linux2 framework. Dit prototype wordt uitgebreid en is een implementatie waarbij er 2 character devices zijn aangemaakt en de huidige Blackmagic driver structuur is aangepast met de nieuwe Video4Linux2 functies. Ook is het mogelijk via een extern programma een I/O control naar de capturekaart te sturen.

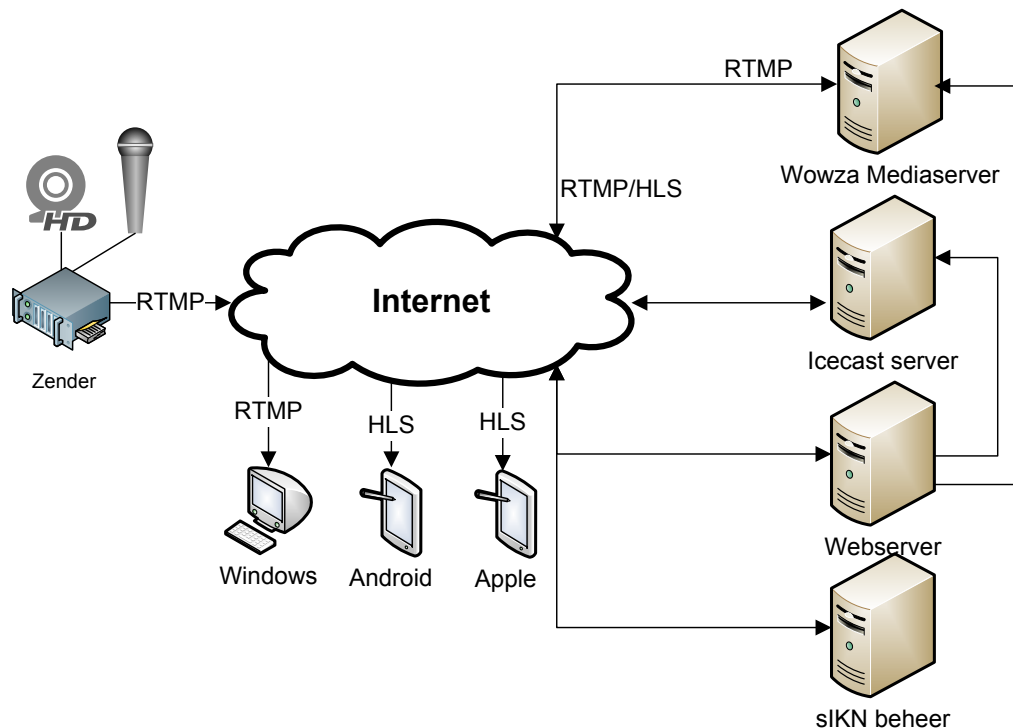
In de `/dev/` structuur zijn het device `blackmagic0` en het device `video0` aangemaakt. Uiteindelijk zal de driver zodanig worden ontwikkeld, dat alleen het character device `video0` wordt geregistreerd.

In dit prototype is nog geen conversie aangebracht tussen de I/O controls van het Video4linux2 framework en de Blackmagic driver.

### 5.3. Huidige architectuur

In deze paragraaf wordt de huidige architectuur van de zender en de driver uiteen gezet. De huidige architectuur is vastgelegd in het architectuurrapport [28]. Het Ontwerpen van een systeemarchitectuur is tevens een competentie en is in het architectuurrapport [28] beschreven.

In paragraaf 3.1 is reeds de opzet gegeven voor Figuur 7. Deze is nu uitgebreid met de protocollen die worden gebruikt voor het ontvangen van de livestream of het uitzenden ervan.



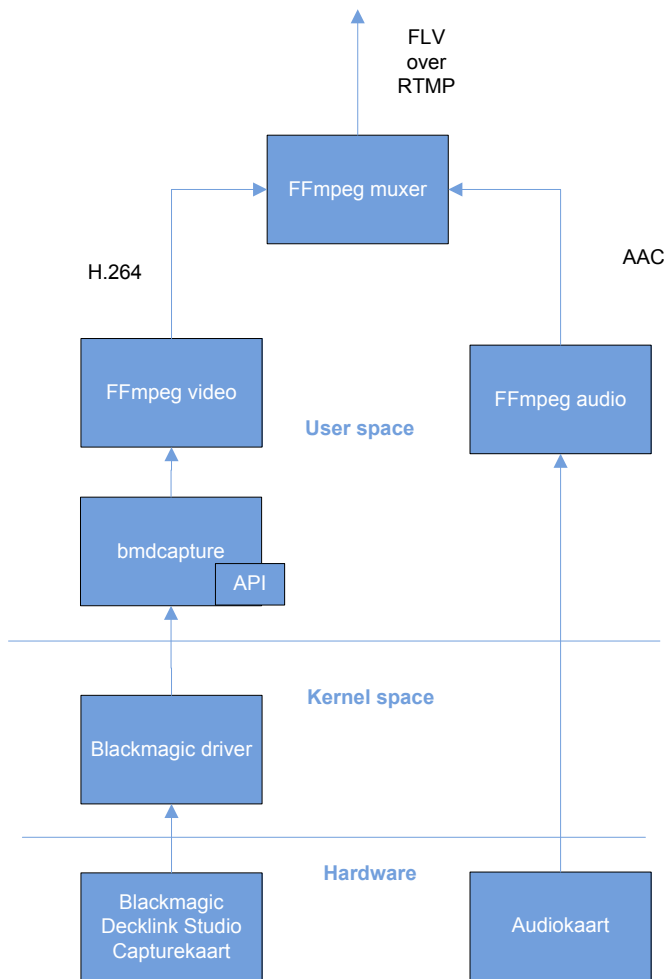
Figuur 7 Globale netwerk infrastructuur

Wanneer de zender gaat capturen wordt door FFmpeg een stream opgezet naar de Wowza Mediaserver. Voor deze stream naar de Wowza server wordt RTMP (Real Time Messaging Protocol) gebruikt. De Wowza server verwerkt de stream en zorgt ervoor dat via diverse apparaten de stream kan worden ontvangen. Afhankelijk van het apparaat wordt RTMP of HLS (HTTP Live Streaming) gebruikt.

Ook is de architectuur van de zender in kaart gebracht. Het betreft hier de hardware die wordt gebruikt voor de zender zoals het moederbord, geheugen, processor maar ook de specificaties van de capturekaart zelf en het besturingssysteem. Het verkrijgen van de hardware specificaties is uitgevoerd door naar de serienummers te kijken die op de chipset zijn aangebracht en geeft een goed beeld van de architectuur en componenten in de zender.

Momenteel wordt op de apparatuur in de kerken gebruik gemaakt van het besturingssysteem Debian Squeeze. Hoewel er een nieuwe versie van Debian genaamd Wheezy beschikbaar is, is er geen noodzaak om dit besturingssysteem te upgraden tijdens de ontwikkeling van deze driver. Dit komt omdat er in Wheezy geen wijzigingen zijn gemaakt die het bereiken van het doel vergemakkelijken. Er is echter wel een nieuwe versie van de x264 codec in Wheezy beschikbaar, deze codec is handmatig voor aanvang van dit project toegevoegd aan het Squeeze besturingssysteem waarmee wordt ontwikkeld en die al reeds in gebruik is.

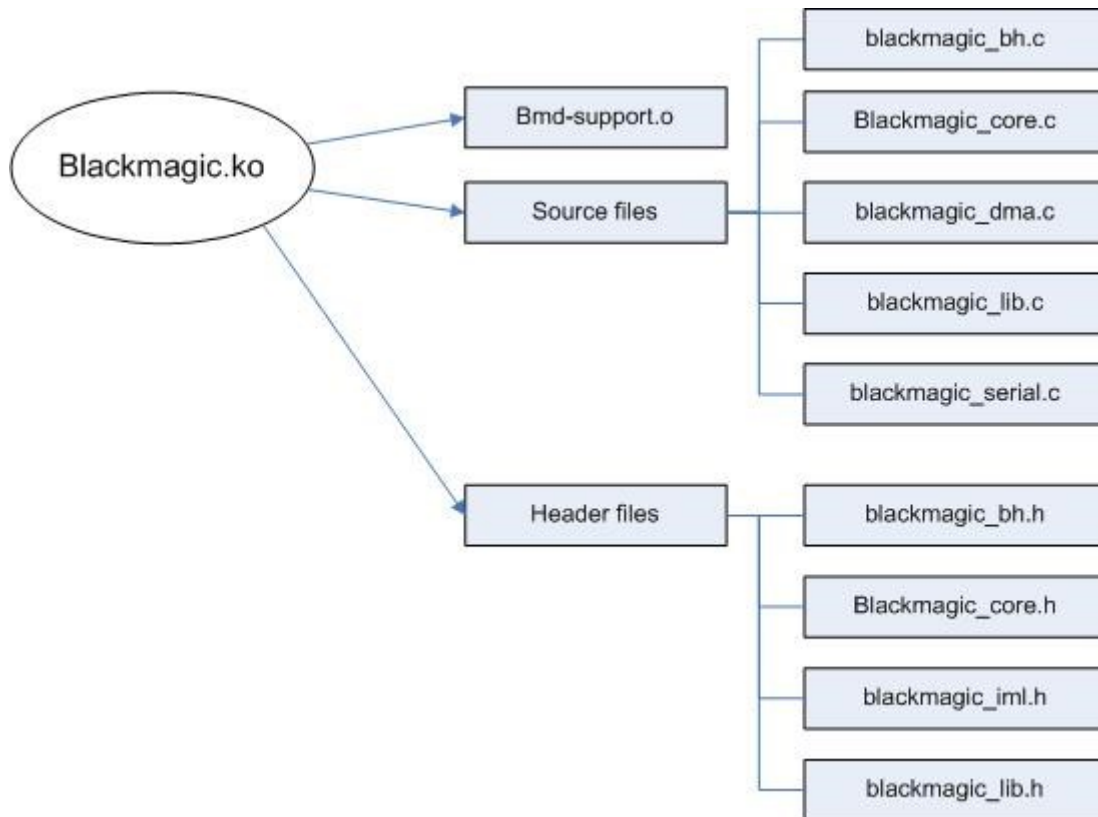
In Figuur 8 is te zien welke stappen momenteel worden uitgevoerd om de data van de capturekaart te streamen naar de servers.



Figuur 8 Huidige werking van de driver

De Blackmagic capturekaart verkrijgt de video content/data die via de camera of mengpaneel wordt doorgegeven. De Blackmagic API verkrijgt via de driver de video data en hiermee wordt deze beschikbaar in het programma bmdcapture. In dit programma bmdcapture wordt een Video4Linux2 header voor de videodata geplaatst waarna het kan worden verwerkt door FFmpeg. Simultaan wordt ook de audio via een andere audiokaart verkregen en doorgegeven aan FFmpeg. FFmpeg voegt hierna de video data en de audio data samen om dit te streamen naar de servers. Het verwerken van audio via het ALSA framework valt buiten de scope van dit project.

Ook is gekeken naar de architectuur van de driver. Wanneer de Blackmagic driver is gecompileerd is het kernel object blackmagic.ko aangemaakt. Uit welke delen de driver bestaat wordt weergegeven in Figuur 9.



Figuur 9 Structuur huidige driver

In Figuur 9 is te zien dat het gecompileerde Blackmagic object bestaat uit een object bdm-support.o een vijftal source files en een viertal header files.

## 5.4. Nieuwe architectuur

In de inceptiefase zijn twee functionele kandidaat systeemeisen beschreven. Deze zijn te vinden in Tabel 1. Ook zijn in deze fase de functionele kandidaat systeemeisen uitgebreid en zullen vanaf hier functionele systeemeisen worden genoemd. Deze functionele eisen zijn weergegeven in Tabel 15. In de elaboratiefase worden ook de niet functionele systeemeisen beschreven [29].

Functionele systeemeisen	
<b>Driver laden</b>	Via modprobe of insmod wordt de driver geladen in de kernel.
<b>Driver instellen</b>	De driver kan via FFmpeg of vanuit het Video4Linux2 framework worden ingesteld. Dit omvat het instellen van de resolutie en het aantal frames per seconde
<b>Video capturen</b>	De driver verkrijgt de video data van de capturekaart
<b>Driver verwijderen</b>	De driver wordt via rmmod verwijderd uit de kernel

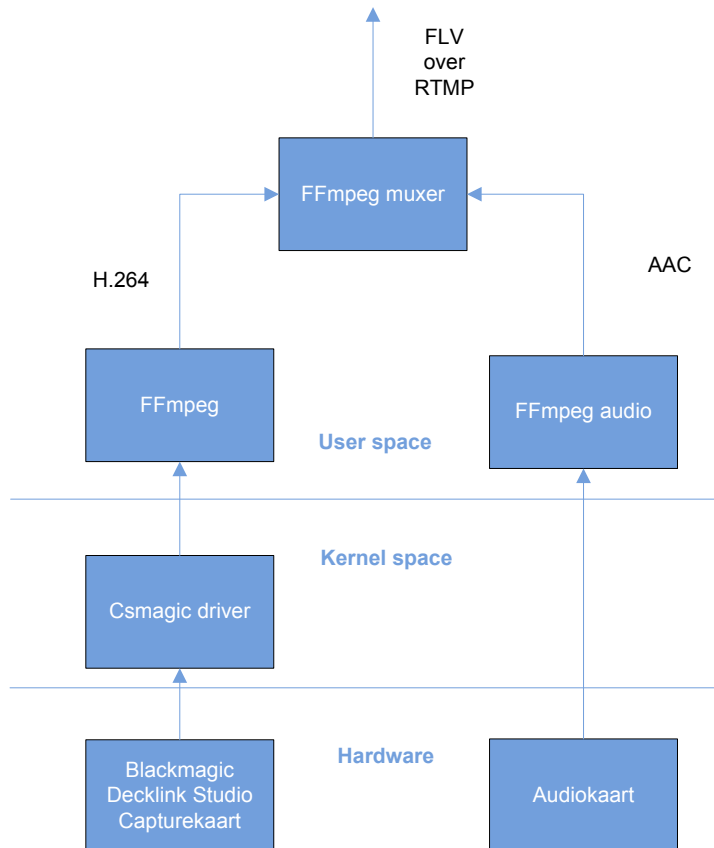
Tabel 15 functionele systeemeisen

Niet functionele systeemeisen specificeren criteria om het systeem te beoordelen maar niet het gedrag zelf [30]. Enkele niet functionele systeemeisen zijn weergegeven in Tabel 16. De volledige lijst is opgenomen in het architectuurrapport [28].

Niet functionele systeemeisen	
<b>De driver moet minimaal compatible zijn met Debian Squeeze</b>	De driver moet compatible zijn met het besturingssysteem Squeeze van Debian. Hogere versies hoeven niet ondersteund te worden, maar indien compatibiliteit met Wheezy kan worden verkregen is dit als toevoeging mooi meegenomen. Lagere versies dan Squeeze hoeven niet te worden ondersteund.
<b>De driver moet compatible zijn met het Video4Linux2 framework</b>	De driver moet compatible zijn met het Video4Linux2 framework. Dit betekent dat er interactie kan plaatsvinden tussen FFmpeg en v4l2-ctl.
<b>De driver moet tenminste ingesteld kunnen worden dat de SDI ingang kan worden gebruikt</b>	De capturekaart bevat diverse ingangen die kunnen worden geselecteerd. Echter ligt de prioriteit bij het verkrijgen van video data via de SDI ingang. Mochten de overige ingangen ook werken, dan is dit mooi meegenomen.

Tabel 16 niet functionele systeemeisen

In Figuur 8 is reeds de huidige situatie van de driver weergegeven. In Figuur 10 wordt de nieuwe situatie voor de driver gegeven. Hierin is te zien dat het programma bmdcapture alsmede de API niet meer aanwezig zijn.



Figuur 10 Nieuwe situatie driver

Door de nieuwe driver is het programma bmdcapture niet meer noodzakelijk omdat FFmpeg direct de video data uit de driver verkrijgt. Hierdoor is het extra transport van data vermeden en worden resources bespaard. FFmpeg zal de video data H.264 gecodeerd doorgegeven naar de FFmpeg muxer waar de video en audio samen worden verwerkt en in een FLV container worden geplaatst waarna het via RTMP naar de server wordt verzonden.

H.264 is een video codec die een zeer sterke compressie van videobeelden nastreeft [31].

FLV is een container die H.264 en AAC ondersteunt [32].

RTMP is een communicatie protocol dat gebruik maakt van het TCP protocol en is ontwikkeld om met hoge performance de data uit te wisselen tussen systemen [33]. Om compatibiliteit te houden met de huidige infrastructuur zal er niet worden afgeweken van de reeds gebruikte protocollen.

Momenteel gebruiken alle zenders die beeld uitzenden RTMP om de video naar Wowza te sturen.

Voordat de capturekaart kan worden gebruikt door FFmpeg dient de capturekaart te worden geïnitieerd. Hiervoor wordt een programma gemaakt genaamd bmd2v4l waarmee de initialisatie van de kaart wordt uitgevoerd. Dit is noodzakelijk omdat voor de initialisatie buffers in user-space worden gebruikt. Hier kan niet van worden afgeweken omdat de blackmagic\_ioctl\_private een copy\_from\_user uitvoert. Hierdoor moet een buffer worden aangemaakt in user-space zodat dit naar de kernel kan worden gekopieerd.

De commando's die uitgevoerd moeten worden zijn eerder al verkregen in Tabel 13. Ook de grootte van de buffer is in deze tabel opgenomen waardoor de initialisatie van de capturekaart kan plaatsvinden.

## 5.5. Ontwikkelaarsrapport

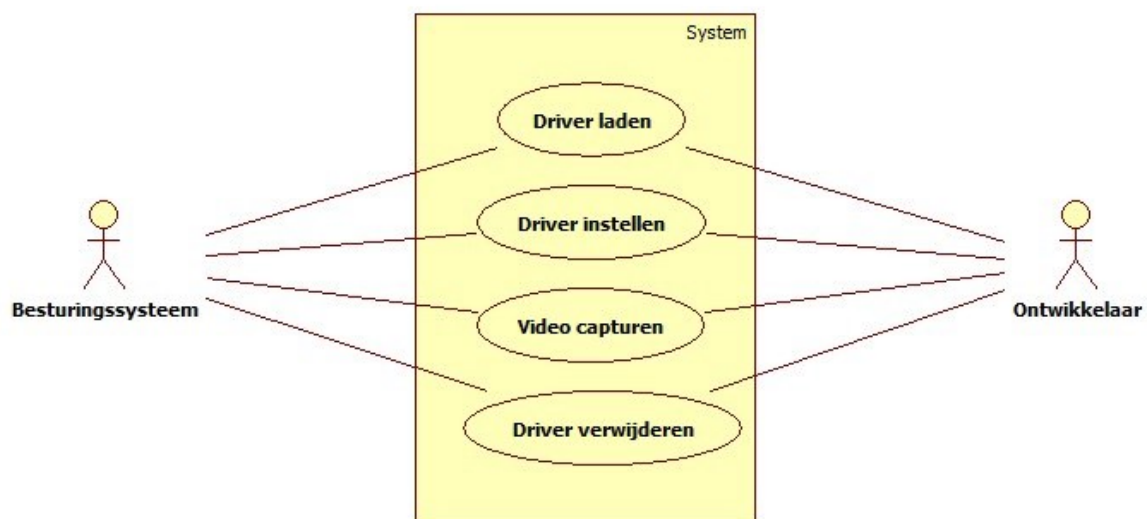
Ook is in de elaboratiefase het ontwerp gemaakt voor de nieuwe driver. Dit is tevens de competentie “Ontwerpen van een technisch informatie systeem” en is vastgelegd in het ontwikkelrapport [34]. Omdat de huidige driver zal worden gebruikt, wordt de nadruk gelegd op het herontwerp van de blackmagic\_core.c. Dit is voor het project voldoende omdat het niet de bedoeling is de gehele driver te vervangen maar wel compatibiliteit te verkrijgen.

Voor het maken van de ontwerpen wordt de modelleertaal UML gebruikt. UML is een standaard voor het voor het maken van ontwerpen voor diverse informatiesystemen. UML kent diagrammen als:

- Klassendiagrammen
- Use-case diagrammen
- Sequentiediagrammen
- Toestandsdiagrammen

UML is echter een taal voor het modelleren van objecten. In de programmeertaal C worden geen objecten gebruikt. Toch kan door middel van UML wel het ontwerp van de driver worden gemaakt. Er is gekozen om de volgende diagrammen te gebruiken: Use-case diagrammen, Sequentie diagrammen, en toestandsdiagrammen [35].

In het vision document [5] is reeds een eerste opzet van de systeemeisen gegeven. Deze systeemeisen zijn in het architectuurrapport uitgebreid [28] en zijn opgenomen in Tabel 15. In Figuur 11 is het volledige use-case diagram getoond.



Figuur 11 Totale use-case

De eerder gemaakte use-case beschrijvingen zijn te vinden in Tabel 2 en Tabel 3. In Tabel 17 en Tabel 18 zijn de overige use-case beschrijvingen voor “driver instellen” en “video capturen” gegeven.

Use case beschrijving	
Naam	Driver instellen
Samenvatting	Door het besturingssysteem of ontwikkelaar wordt de driver ingesteld. Dit is bijvoorbeeld de resolutie en de pixelformaat.
Actoren	Besturingssysteem, ontwikkelaar
Aannamen	<ul style="list-style-type: none"> <li>De zender is gestart.</li> <li>De driver is geladen.</li> <li>De capturekaart is herkend.</li> <li>FFmpeg of v4l2-ctl is op het systeem aanwezig.</li> <li>De actor ontwikkelaar heeft beschikking over een terminal</li> </ul>
Beschrijving	<ol style="list-style-type: none"> <li>De actor geeft aan FFMpeg of v4l2-ctl de parameters mee voor de driver.</li> <li>Wanneer de parameters in de driver zijn opgeslagen retourneert de v4l2-ctl of FFMpeg de waarde 0. Lukt dit niet treedt uitzondering [ foutieve parameters ] op.</li> </ol>
Uitzonderingen	<ul style="list-style-type: none"> <li>[ foutieve parameters ] Er is een ongeldige combinatie voor de configuratie opgegeven. De waarde 1 wordt geretourneerd.</li> </ul>
Resultaat	De driver is ingesteld met de opgegeven parameters.

Tabel 17 use-case driver instellen

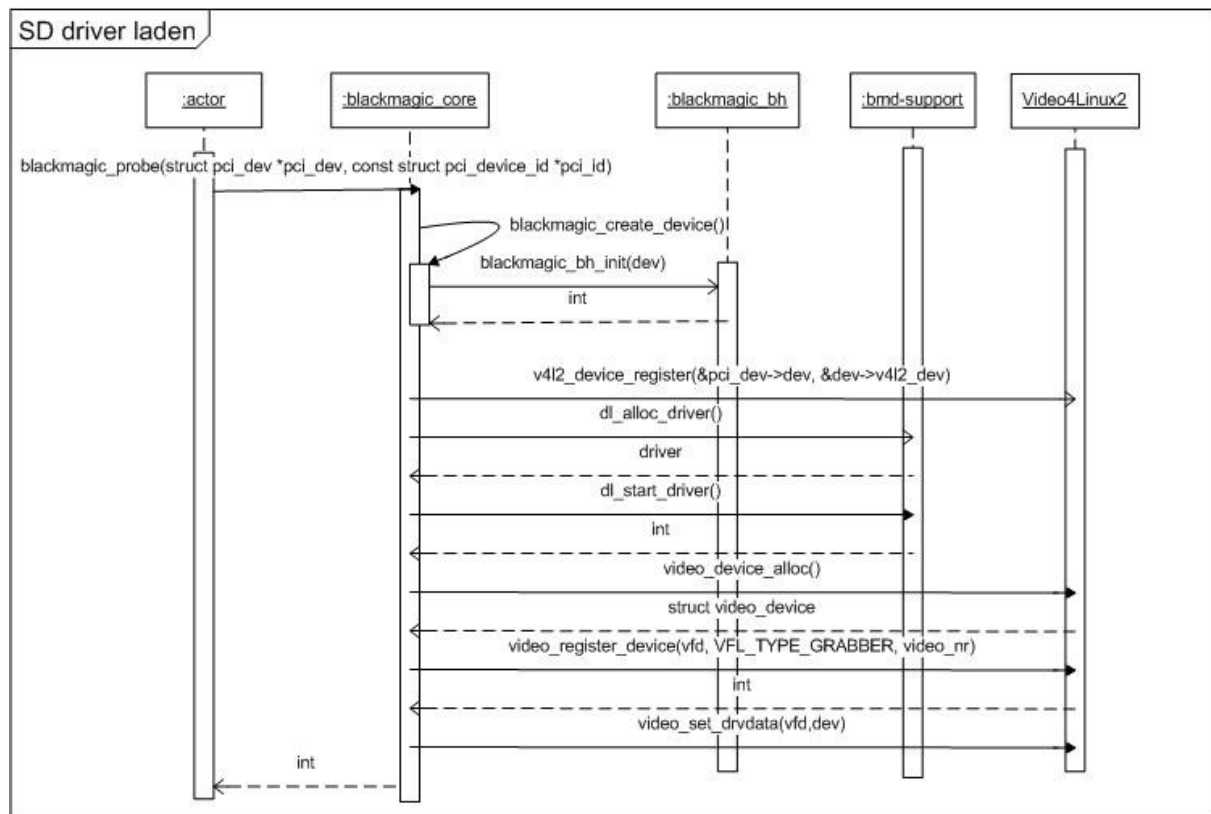
Use case beschrijving	
Naam	Video captureren
Samenvatting	De actor besturingssysteem of ontwikkelaar zal de videocapture mode activeren.
Actoren	Besturingssysteem, Ontwikkelaar
Aannamen	<ul style="list-style-type: none"> <li>De zender is gestart.</li> <li>De driver is geladen.</li> <li>De capturekaart is herkend.</li> <li>FFmpeg of v4l2-ctl is op het systeem aanwezig.</li> <li>De actor ontwikkelaar heeft beschikking over een terminal</li> <li>De driver is reeds ingesteld met de gewenste parameters</li> </ul>
Beschrijving	<ol style="list-style-type: none"> <li>De actor geeft het commando aan FFMpeg of v4l2-ctl om in streaming mode te gaan.</li> <li>De capturekaart begint het captureren van de video data. FFMpeg of v4l2-ctl retourneert 0 wanneer de stream is gestart. Wanneer dit niet lukt, treedt een uitzondering op [ Kan streamen niet starten ]</li> </ol>
Uitzonderingen	<ul style="list-style-type: none"> <li>[ Kan streamen niet starten ] De driver kan niet streamen omdat de configuratie onjuist is en retourneert de waarde 255.</li> </ul>
Resultaat	De applicatie FFMpeg ontvangt streaming videodata.

Tabel 18 use-case Video captureren



Om de flow aan te geven welke functies van de driver worden aangeroepen zijn er sequentie diagrammen gemaakt. Hiermee wordt een goed beeld geschetst van wat er gebeurt tijdens bijvoorbeeld het laden van de driver. In het sequentiediagram Figuur 12 wordt de blackmagic\_core.c benaderd als object. Hier is voor gekozen omdat wanneer de blackmagic\_core.c is gecompileerd, dit een object is geworden. Door dit uitgangspunt wordt de filosofie van het UML sequentiediagram intact gelaten.

In Figuur 12 is het sequentiediagram gegeven voor de sequentie driver laden. Hier zijn de diverse functie calls te zien die tijdens het laden van de driver worden gedaan.



Figuur 12 sequentie diagram driver laden

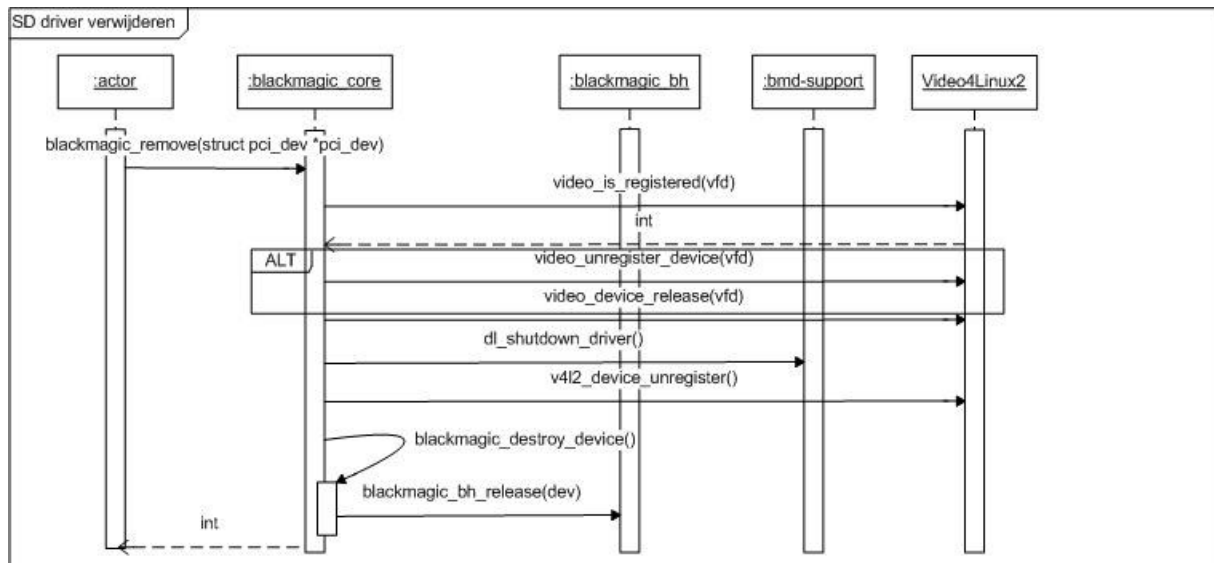
In Figuur 12 is de sequentie weergegeven voor het laden van de driver. In dit diagram is te zien welke functies er worden aangeroepen tijdens het laden van de driver. De functie blackmagic\_probe binnen de blackmagic\_core wordt aangeroepen. Hierdoor wordt de functie blackmagic\_create\_device aangeroepen binnen de blackmagic\_core. Hier wordt in het blackmagic\_bh object de functie blackmagic\_bh\_init aangeroepen waarna binnen het Video4Linux2 framework de functie v4l2\_register\_device wordt aangeroepen.

Vervolgens worden binnen bmd-support de functie dl\_alloc\_driver en dl\_start\_driver uitgevoerd. Via de functie video\_device\_alloc wordt door het video4linux2 framework een struct video\_device geretourneerd.

Door video\_register\_device aan te roepen, wordt aangegeven wat voor type device er aangemaakt moet worden. De parameter vfd is de struct video\_device, de parameter type\_grabber geeft aan wat voor type device het is en het video\_nr kan worden gebruikt om handmatig een minor nummer toe te kennen aan het video device.

Door de functie `video_set_drvdata` wordt de nieuwe private data aan het device toegevoegd. Hierna is de registratie van het nieuwe video device afgerond en is deze klaar voor gebruik.

In Figuur 13 wordt het sequentiediagram voor het verwijderen van de driver getoond.



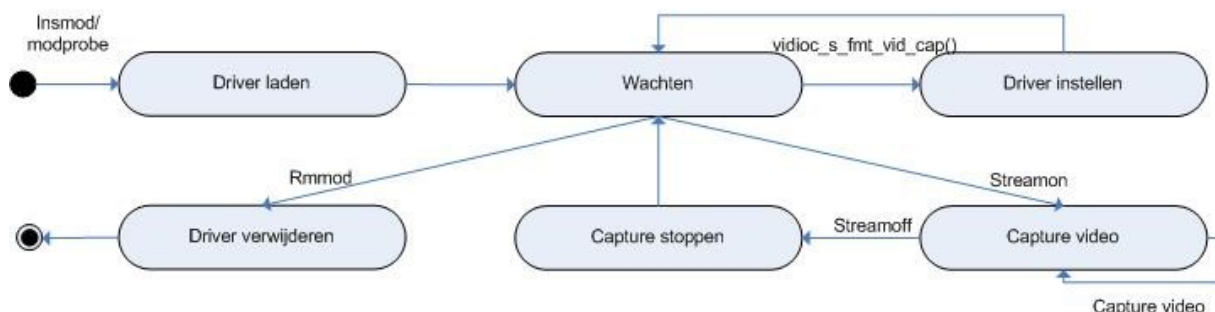
Figuur 13 Driver verwijderen

De actor roept via `rmmod` de functie `blackmagic_remove` in de `blackmagic_core` aan. In deze functie wordt gekeken of het `video_device` inderdaad is geregistreerd. Wanneer dit het geval is worden de functies in de alt sectie uitgevoerd. Door de aanroep van `video_unregister_device` en `video_device_release` wordt het daadwerkelijke character device verwijderd.

Hierna wordt de functie `dl_shutdown_driver` aangeroepen binnen de `bmd-support`, waarna `v4l2_device_unregister` wordt uitgevoerd. Hiermee worden eventuele subdevices opgeruimd, het gereserveerde geheugen aan de kernel terug gegeven en wordt de privatedata van het `Video4Linux2` device `NULL` gemaakt.

Via de functie `blackmagic_destory_device` wordt de bottomhalf handler gereleased. Dit gebeurt door de functie `blackmagic_bh_release` in de `blackmagic_bh` aan te roepen.

Ook is er in het ontwikkelrapport een toestandsdiagram voor de driver opgesteld. In Figuur 14 is weergegeven in welke toestanden de driver zich kan bevinden.



Figuur 14 Toestand diagram driver

Hieronder wordt de beschrijving gegeven voor Figuur 14.

1. Via insmod of modprobe wordt de driver geladen.
2. De driver wacht vervolgens tot hij in de streaming mode wordt gezet, wordt verwijderd of ingesteld.
3. Wanneer via streamon het capturen wordt gestart zal er data worden gecaptured.
4. De driver blijft in de capture mode totdat de capture wordt gestopt.
5. Wanneer de capture is gestopt zal de driver wachten tot het opnieuw moet capturen of de driver wordt verwijderd.
6. Wanneer het capturen wordt hervat wordt de cyclus herhaald vanaf punt 3.
7. Wordt de driver verwijderd, dan wordt de driver uit de kernel verwijderd.

## 6. Constructiefase

De huidige driver wordt zoals eerder is aangegeven compatible gemaakt met het Video4Linux2 framework. Dit is noodzakelijk zodat FFmpeg dadelijk direct het device kan openen zonder dat er een conversie in user space uitgevoerd hoeft te worden. Voor het maken van de csmagic driver is gekeken naar de functies die in het Video4Linux framework aanwezig zijn.

Ook is er gekeken naar voorbeelden van andere Video4Linux2 device drivers. Voorbeelden hiervan zijn: cx23558 en vivi.c. Dit zijn open source projecten waaruit informatie over de werking van het Video4Linux2 framework kan worden gehaald, maar waar ook informatie voor de ontwerpen uit is gehaald. Deze source bestanden zijn te vinden in de Linux source files die uit de Debian repository kunnen worden gedownload met het commando `"apt-get install linux-source"`. De voorbeeld bestanden bevinden zich in de map `"{source folder}/drivers/media/video/"`. In deze map staan ook de Video4Linux2 sources files zoals v4l2-dev.c waarin functies als `video_device_alloc()` te vinden zijn. De header files kunnen worden gevonden op de locatie `"{source folder}/include/"`.

De huidige Blackmagic driver maakt een character device blackmagic0 aan. Deze driver bevat geen Video4Linux2 functies. Omdat de nieuwe driver volledig Video4Linux2 compatible moet zijn, is de huidige driver aangepast zodat deze een character device `/dev/video0` aanmaakt met major nummer 81. Hiervoor is het eerder gemaakte ontwerp uit Figuur 12 gebruikt.

Om het Video4Linux2 device aan te maken dienen in de blackmagic\_probe bijvoorbeeld de functies te worden toegevoegd die zijn weergegeven in Tabel 19.

Functie	Beschrijving
<b>v4l2_device_register</b>	Registreert het Video4Linux2 device
<b>video_device_alloc</b>	Retourneert de Video4Linux device struct
<b>csmagic_template</b>	Bevat de video_device operaties <ul style="list-style-type: none"> <li>• fops</li> <li>• ioctl_fops</li> <li>• release</li> </ul>
<b>video_register_device</b>	is een belangrijke omdat hiermee het character device wordt aangemaakt. Dit is dan in <code>/dev</code> terug te vinden als bijvoorbeeld <code>"81 0 video0"</code> . Hierbij is 81 het major nummer, 0 is het dynamisch toegekende minor nummer en video0 is de naam van het character device.
<b>video_set_drvdata</b>	Voegt de Video4Linux2 private data toe aan het Blackmagic device.

Tabel 19 toegevoegde functies aan de blackmagic\_probe

Nadat de functie blackmagic\_probe is afgerond is het Video4Linux2 device aangemaakt. Echter moeten hier nog operaties aan worden meegegeven zodat device gegevens kunnen worden opgevraagd.

De huidige Blackmagic file operations zoals te zien zijn in Figuur 4, wordt vervangen door de Video4Linux2 device operations. Door het toevoegen van de nieuwe device operations moeten ook de parameters van de aanroepende functie worden aangepast, omdat bijvoorbeeld door het Video4Linux2 framework vaak als ingaande parameter niet de struct file wordt gebruikt terwijl Blackmagic hier veelal wel gebruik van maakt. Deze incompatibiliteit dient op diverse plaatsen verholpen te worden. Anders bestaat het risico dat het systeem onverwacht/ongewenst gedrag gaat vertonen.

Omdat de capturekaart door de omzetting naar het Video4Linux2 framework niet kan worden geïnitieerd vanuit de API is een programma genaamd bmd2v4l geschreven waarmee de I/O controls kunnen worden gestuurd naar de driver. De reden dat dit vanuit een extern programma gebeurt, is omdat vanuit de bmd-support een copy\_from\_user of copy\_to\_user wordt uitgevoerd. Met deze functie wordt een user buffer gekopieerd naar de kernel-space of een kernel buffer naar user-space gebracht. Zou de buffer dus niet uit user-space komen, dan gaat dit problemen opleveren. Ook wordt dit gebruikt om aan te tonen dat van buiten de driver handmatig I/O controls kunnen worden verwerkt. Hiervoor moet dus een buffer worden aangemaakt in user-space met een specifieke grootte en worden meegegeven aan de functie ioctl met de bijbehorende waarde. In Tabel 13 is te zien dat de informatie over de buffer grootte reeds is verkregen. Hiermee kan de initialisatie worden uitgevoerd van de capturekaart.

Omdat het programma bmd2v4l geen gebruik maakt van DMA of data transporteert, heeft deze geen invloed op de performance van de zender. Ook moet de driver dusdanig functioneren dat deze ongeacht vanuit welk programma het wordt geopend de instellingen moet onthouden.

Het sturen van een I/O control wordt gedaan door het character device video0 te openen waarmee de device operations en file operations van de driver beschikbaar worden. Door het openen van het character device wordt een file handle verkregen die kan worden gebruikt om een I/O control naar dit device te sturen.

Het sturen van een I/O control naar de driver wordt vervolgens gedaan door de functie *ioctl* aan te roepen vanuit het programma.

Bij het ontwikkelen van de driver komt ook debugging kijken. Debugging moet worden uitgevoerd wanneer bijvoorbeeld een kernel panic ontstaat of wanneer er ongewenst gedrag van het systeem optreedt. Veelal wordt dit veroorzaakt door te schrijven naar een geheugen locatie die te klein is voor de te schrijven data, of omdat een geheugen locatie reeds is vrijgegeven. Een kernel panic is een actie van het besturingssysteem wanneer er een fatale fout is opgetreden die niet hersteld kan worden [36]. De error met de stacktrace wordt dan op het scherm getoond. Soms is de fout dermate fataal dat er geen output over de fout te vinden is. In Voorbeeld 3 is een deel van een kernel panic te zien die is ontstaan tijdens het testen van de driver.

```
BUG: unable to handle kernel paging request at f9650260
IP: [<c12c3c68>] mutex_lock+0xb/0x21
*pdpt = 0000000001486001 *pde = 0000000033760067 *pte = 800000011be57161
Oops: 0003 [#1] SMP

EIP: 0060:[<c12c3c68>] EFLAGS: 00010246 CPU: 7
EIP is at mutex_lock+0xb/0x21
EAX: f9650260 EBX: f9650260 ECX: f3b19800 EDX: 00000000
ESI: f2eb3ad0 EDI: f2c06c00 EBP: 00000000 ESP: f2eb3a10
```

*Voorbeeld 3 Kernel panic*

Het niet meer reageren van het besturingssysteem wordt ook wel een freeze genoemd. Hierdoor is het met regelmaat voorgekomen dat de zender via een power cycle opnieuw moest worden gestart in verband met de kernel freeze/panic. Tijdens het opstarten van het besturingssysteem worden ook de kernel modules geladen. Doordat het systeem niet meer reageerde, was het niet mogelijk de kernel module te verwijderen. Hierdoor werd de module tijdens het opstarten weer geladen waarna wederom een panic kon optreden.

Hierop wordt echter geanticipeerd door een tweede kernel te installeren. Door het starten van de tweede kernel kon de module worden verwijderd, het systeem opnieuw gestart en kon worden uitgezocht waardoor de kernel panic was ontstaan.

Het achterhalen waardoor een kernel panic ontstaat is veelal een complexe handeling. Dit kan bijvoorbeeld worden gedaan door een object dump te maken van het totale gecompileerde kernel object csmagic en de locatie op te zoeken in disassembly waar het mis is gegaan. Hiermee kan worden achterhaald waar het mis is gegaan.

Meestal heb ik ervoor gekozen om te debuggen door het toevoegen van logregels. Hiermee kan snel in de code achterhaald worden op welke locatie de fout is opgetreden. Het gevaar van deze methode is echter wel dat een kernel panic soms pas later ontstaat dan de regel die in de log is geschreven, waardoor in de verkeerde hoek gezocht kan worden.

Een voordeel van iteratief werken is tevens dat er steeds kleine delen worden ontwikkeld waardoor de scope waarbinnen een fout kan optreden beperkt blijft.

## 6.1. Testen

In het ontwikkelrapport [34] zijn ook de diverse test scenario's beschreven. Het testen van een softwaresysteem is tevens een competentie. Een uitgangspunt voor testen is dat testen reproduceerbaar zijn en dat ze een specifieke systeemeis beschrijven. Dit betekent dat zonder wijzigingen aan de driver, de resultaten van de testen hetzelfde zijn, ongeacht hoe vaak deze worden uitgevoerd. In dit testplan zijn black-box testen niet opgenomen omdat hiervoor binnen de organisatie reeds testplannen aanwezig zijn. Voor de testen die tijdens het ontwikkelen worden uitgevoerd zijn wel test scenario's uitgewerkt. Via het testen wordt de betrouwbaarheid en de functionaliteit van de driver getest. De testen die uitgevoerd worden zijn white-box testen, waarmee de gestelde eisen aan het systeem getest kunnen worden.

Omdat voor elke keer dat een test wordt uitgevoerd, de uitgangspunten hetzelfde moeten zijn, worden er aan een test randvoorwaarden gesteld.

Door het uitvoeren van de onderstaande test wordt de functionele eis "Driver laden" getest. Ook worden hiermee enkele niet functionele eisen getoetst. De te toetsen eisen voor deze test zijn:

- De driver moet minimaal compatible zijn met Debian Squeeze
- De driver moet compatible zijn met het Video4Linux2 framework

De niet functionele eisen worden getoetst doordat de driver kan worden geladen en er een video device met major nummer 81 wordt aangemaakt. Uiteraard worden met deze test niet de volledige niet functionele eisen getest maar enkel een gedeelte hiervan.

De test condities voor de test "driver laden" met de bijbehorende omschrijving is te vinden in Tabel 20. Deze testen kunnen worden uitgevoerd als aan de randvoorwaarden die hieronder zijn opgesteld is voldaan. In dit hoofdstuk is echter maar één test opgenomen. De overige testen zijn te vinden in het ontwikkelrapport [34].

## Randvoorwaarden

- De zender is ingeschakeld
- De kernel module is nog niet geladen
- Het kernel object csmagic is aanwezig
- De driver dient te worden geladen met root rechten
- De uit te voeren commando's worden vanaf de terminal uitgevoerd.

## Test

Test conditie	Omschrijving	Geslaagd	Gefaald
De driver kan worden geladen in de kernel?	Dit kan worden gedaan door insmod csmagic.ko uit te voeren vanuit de map waar dit kernel object te vinden is.		
Is de driver in de kernel geladen?	Dit kan worden gecontroleerd door lsmod   grep csmagic uit te voeren		
Is er een video device aangemaakt?	Dit kan worden gecontroleerd door ls -l /dev/   grep video uit te voeren. Hier moet een video0 te vinden zijn		
Heeft de video0 het juiste major nummer?	Het major nummer voor video0 moet zijn 81. Bijvoorbeeld: 81, 0 May 2 09:17 video0		
Kan de module informatie worden geraadpleegd?	Dit kan worden gedaan door modinfo csmagic uit te voeren.		
Staat bij de filename de locatie waarvandaan de module is geladen?	Wanneer het script ./go.sh wordt uitgevoerd zal hier staan: /lib/modules/3.2.0-4-686-pae/kernel/drivers/media/video/csmagic.ko.		
De version komt overeen met de versie van de driver?	De versie van de driver is terug te vinden in de blackmagic_core.c		
Staat bij description de beschrijving van de driver?	Deze moet overeen komen met de beschrijving in de blackmagic_core.c		
Staat bij author de auteur vermeld?	Dit is de author zoals te vinden is in de blackmagic_core.c		
Blijft het systeem stabiel na het laden van de driver?	Dit kan worden gecontroleerd door bijvoorbeeld een toets op het toetsenbord in te drukken of de driver een tijd geladen te houden.		
Zijn er in de system log fouten te zien gerelateerd aan de driver?	De meldingen van de driver zijn te vinden in /var/log/syslog. Elke log regel heeft de waarde blackmagic: in zich.		
<b>Resultaat</b>			

Tabel 20 Test driver laden

## 7. Evaluatie

In dit hoofdstuk wordt het project geëvalueerd. Dit hoofdstuk is verdeeld in twee paragrafen, namelijk de product evaluatie en de proces evaluatie. In de product evaluatie worden de op te leveren producten geëvalueerd terwijl de proces evaluatie het algemene proces beschrijft.

### 7.1.Product evaluatie

In deze paragraaf worden de op te leveren producten geëvalueerd.

#### Vision document

Het vision document bevat de opzet van het project. Zo worden hier een aanleiding, probleemstelling en een doelstelling geformuleerd. Maar wordt ook de scope van het project bepaald, een planning gemaakt en de risico's in kaart gebracht.

Doordat de probleemstelling voor mij nog niet helemaal duidelijk was aan het begin van het project heeft het redelijk wat tijd gekost deze opdracht goed te beschrijven en te voorzien wat nu daadwerkelijk het probleem was. Dit kwam voornamelijk doordat de noodzakelijke achtergrond kennis, aan het begin van het project, bij mij ontbrak. Toch ben ik erin geslaagd een aanleiding, probleemstelling en doelstelling te formuleren die aansluiten bij de verwachtingen van de opdrachtgever.

Ook zijn er in het vision document kandidaat systeemeisen opgesteld. Het zijn kandidaat systeemeisen, omdat deze pas definitief worden nadat de reverse engineering is uitgevoerd. Tijdens de reverse engineering worden namelijk de daadwerkelijke eisen die aan het systeem gesteld worden achterhaald. Deze kandidaat systeemeisen zijn noodzakelijk om een eerste use-case op te kunnen stellen.

Het uitvoeren van een risico analyse is complex, omdat het niet eenvoudig is vooraf te bepalen wat het slagen van het project in gevaar kan brengen. Ook moet voor het analyseren van de risico's de complete scope van het project verkend worden.

In het vision document is een risico analyse gemaakt dat de project risico's afdekt. Twee van de risico's zijn opgenomen in hoofdstuk 4 in Tabel 7 en Tabel 8.



## Analyserapport

In het analyserapport wordt een beschrijving gegeven hoe de reverse engineering is uitgevoerd en wat de verkregen resultaten zijn.

De beschrijving van de reverse engineering is te vinden in paragraaf 5.1.

Het opstellen van het volledige analyserapport heeft veel tijd gekost. Dit komt doordat ik bij aanvang van dit project niet beschikte over de benodigde voorkennis voor het uitvoeren van de reverse engineering. Het gaat dan bijvoorbeeld om technieken zoals het uitvoeren van de reverse engineering, het lezen van assemblercode maar ook het gebruiken van het programma objdump en het analyseren van de driver. Wel ben ik simultaan aan het architectuurrapport gaan schrijven. Het beschrijven van de architectuur in het architectuurrapport kon simultaan worden uitgevoerd, omdat tijdens de reverse engineering ook de hardware van de zender in kaart kon worden gebracht.

Het toevoegen van de log regels in de `blackmagic_ioctl` heeft bijgedragen aan het verkrijgen van de I/O controls die worden gebruikt voor de aansturing van de capturekaart. De toevoeging van de log regels heeft echter een aantal keren een kernel panic ten gevolge gehad. Zeker aan het begin van dit project is het een lastige klus gebleken te achterhalen waardoor dit werd veroorzaakt. Ook het debuggen van de driver heeft uiteindelijk een grote impact gehad op het project.

Het splitsen van het programma Capture heeft veel inzicht gegeven in de werking van de capturekaart en de API. Deze splitsing is noodzakelijk geweest voor het verkrijgen van de waarden behorend bij een I/O control. Onder deze waarden kan worden verstaan de diverse displaymodes die door de capturekaart worden ondersteund. Deze displaywaarden zijn vastgelegd in het analyserapport waarvan een gedeelte is weergegeven in Tabel 10.

Tijdens het uitvoeren van de reverse engineering heb ik diverse nieuwe technieken geleerd. Eén van deze technieken omvat het maken en lezen van een object dump waarmee de functienamen, behorend bij een I/O control, kunnen worden achterhaald. Ook is de huidige Blackmagic driver geanalyseerd en heb ik mij verdiept in de kernel functies die in deze driver worden gebruikt.

Het linken van de I/O controls en hun waarden aan de API functies is tijdrovend geweest. Dit komt doordat er dermate veel regels in de log werden geschreven waardoor niet te achterhalen was welke I/O control bij een API functie behoorde. Hiervoor heb ik de assemblercode geraadpleegd. Het analyseren van de assemblercode is echter een complexe opgave maar heeft er wel toe geleid dat de functienaam aan de bijbehorende I/O control konden worden gelinkt.

De door reverse engineering verkregen resultaten heb ik samengevat in het analyserapport [7] waarvan een deel is opgenomen in Tabel 13 van dit document. In deze tabel zijn ook de functienamen vastgelegd die via de objectdump zijn verkregen. Deze resultaten kunnen worden gebruikt om de Video4Linux2 I/O controls via een mapping door te sturen naar de functie `blackmagic_ioctl`. Door de Video4Linux2 I/O controls te vervangen door de bijbehorende Blackmagic I/O control kan de Blackmagic DeckLink Studio capturekaart worden aangestuurd.

## Architectuurrapport

In het architectuurrapport is de huidige architectuur gedocumenteerd. Een groot deel van de huidige architectuur is verkregen tijdens het uitvoeren van reverse engineering. De huidige architectuur is vastgelegd in het architectuurrapport [28]. Tijdens de reverse engineering is het namelijk noodzakelijk geweest te bepalen wat de specificaties zijn van bijvoorbeeld de Blackmagic DeckLink Studio capturekaart. In dit document heb ik tevens beschreven hoe het gehele proces van het capturen van video tot het afspelen ervan op dit moment plaatsvindt. Hiermee zijn de architectuur en de infrastructuur in kaart gebracht. Ook is hiermee vastgelegd welke protocollen worden gebruikt tussen de diverse systemen binnen DCT Net.

Ook is gekeken naar het huidige besturingssysteem Debian Squeeze en het nieuwe besturingssysteem Debian Wheezy om te achterhalen of er een noodzaak was het huidige besturingssysteem te upgraden. Omdat er geen changes in Wheezy zijn gemaakt die het te bereiken doel vergemakkelijken, is besloten van het huidige besturingssysteem Squeeze gebruik te blijven maken.

Naar aanleiding van de reverse engineering heb ik de functionele en niet functionele eisen voor de nieuwe csmagic driver in kaart gebracht. Hiermee zijn de bestaande functionele kandidaat systeemeisen die zijn gedefinieerd in het vision document [5] omgezet naar functionele systeemeisen en zijn deze uitgebreid. Deze functionele en niet functionele systeemeisen dienen als input voor het ontwikkelrapport waar de ontwerpen worden gemaakt voor de nieuwe driver.

## Ontwikkeldocument

In het ontwikkelrapport is het ontwerp gemaakt voor de nieuwe csmagic driver. Zoals eerder aangegeven wordt hierbij gebruik gemaakt van de taal UML. Hoewel UML is toegespitst op object georiënteerd ontwikkelen, heb ik er toch voor gekozen deze methodiek te gebruiken. Overigens heb ik geen methodiek gevonden die beter aansluit bij het ontwikkelen van een driver. Voor het ontwerpen van de driver heb ik gebruik gemaakt van use-case diagrammen, sequentie diagrammen en een toestandsdiagram [35].

Om de flow van bijvoorbeeld het starten van de driver weer te geven is dit vastgelegd in een sequentie diagram. Ik heb gekozen om het als een object te benaderen omdat een source file na het compileren een object is geworden. Hiermee blijven de technieken voor het maken van een sequentie diagram intact en de filosofie van UML behouden.

Tijdens het project heb ik test scenario's geschreven om de driver tijdens het ontwikkelen te kunnen testen. Deze testen zijn white-box testen en derhalve toegespitst op het testen van de driver tijdens de ontwikkelingsfase. Er zijn testplannen in de organisatie aanwezig waarmee de gebruikerservaring en functionaliteit kan worden getest door middel van een black-box test. De ontwikkelde white-box test scenario's zijn te vinden in het ontwikkelrapport [34].

Ik heb ervoor gekozen de scenario's dusdanig te schrijven dat het laden van een driver een scenario op zich is. Daarnaast is bijvoorbeeld het instellen van de driver ook een aparte test. Hierdoor kunnen delen van het systeem worden getest maar ook het systeem als geheel. Ook heb ik de test condities geschreven zodat hier enkel een eenduidig antwoord op kan komen namelijk: aan de test conditie is voldaan of aan de test conditie is niet voldaan. Ik heb ook de test scenario's herhaalbaar gemaakt door bij elk scenario randvoorwaarden op te nemen. De manier waarop de testen momenteel zijn geformuleerd sluiten tevens aan bij de formulering binnen de organisatie.

## 7.2. Proces evaluatie

Vanuit mijn werk heb ik 2 dagen (16 uur) in de week beschikbaar gekregen om aan het afstuderen te besteden. De overige studie uren heb ik naast mijn werk, in mijn vrije tijd gedaan. De uren die ik op mijn werk heb kunnen besteden ben ik voornamelijk bezig geweest met het uitvoeren van reverse engineering en het omzetten van de driver naar het Video4Linux2 framework.

Bij aanvang van het project ben ik mij eerst gaan verdiepen in de materie. Het betreft hier het verkrijgen van kennis over C, Linux, Drivers en het Video4Linux2 framework. Het verkrijgen van deze kennis heeft me echter meer tijd gekost dan ik had gepland, maar het heeft een positief effect gehad op het uitvoeren van de reverse engineering en het ontwikkelen van de driver.

Naarmate het project vorderde was ik beter in staat de kernel en driver functies in hun context te plaatsen. Zo werd voor mij duidelijk waarom de driver niet te benaderen was vanuit FFmpeg en ook kreeg ik inzicht in hoe een driver werkt en welke stappen er worden doorlopen om daadwerkelijke de driver te kunnen gebruiken binnen het Video4Linux2 framework. Deze kennis is uit diverse bronnen verkregen [37] [26] [38].

Tijdens het maken van de driver ben ik veelvuldig geconfronteerd met een kernel panic. Het oplossen en achterhalen van de fouten die dit veroorzaken is veelal tijdrovend, omdat aan de hand van de stacktrace (als die er al is) achterhaald moet worden bij welke instructie de fout is opgetreden. Hierna moet nog onderzocht worden wat de kernel panic heeft veroorzaakt. Dit kan namelijk verschillende oorzaken hebben zoals het schrijven naar delen in het geheugen die niet door de driver zijn gereserveerd of door een foutieve typecasting binnen de driver. Daarom heb ik vooral eigen logregels in de driver toegevoegd waarbij veel sneller een mogelijke oorzaak kan worden gevonden.

Dit project was voor mij een grote uitdaging omdat ik mij in veel diverse aspecten heb moeten verdiepen zoals eerder is aangegeven. Doordat ik mij heb moeten verdiepen in onbekende gebieden heb ik mijn algemene kennis over het uitvoeren van reverse engineering en het ontwikkelen van een driver verrijkt.

Hoewel de driver nog in ontwikkeling is, heb ik kunnen vaststellen dat het zeker haalbaar is, een werkende Video4Linux2 driver te maken die de Blackmagic DeckLink Studio capturekaart kan aansturen. Dit is aangetoond doordat de initialisatie van de capturekaart vanuit een extern programma genaamd bmd2v4l kon worden uitgevoerd.

## Bibliografie

- [1] Kerkomroep, „Kerkomroep,” DCT Net, [Online]. Available: [www.kerkomroep.nl](http://www.kerkomroep.nl).
- [2] Kerkomroep, „Android Play store,” [Online]. Available: <https://play.google.com/store/apps/details?id=corp.csnet.nl.kerkomroep&hl=nl>.
- [3] Kerkomroep, „Kerkomroep Itunes,” [Online]. Available: <https://itunes.apple.com/nl/app/kerkomroep/id541059718?mt=8>.
- [4] J. Warmer en A. Kleppe, „Use-case-diagram,” in *Praktisch UML*, Amsterdam, Pearson Education Benelux bv, 2008, pp. 85 - 102.
- [5] M. Hamburg, „Visiondocument,” 2014.
- [6] Wikipedia, „Reverse engineering,” Wikipedia, 10 10 2013. [Online]. Available: [http://nl.wikipedia.org/wiki/Reverse\\_engineering](http://nl.wikipedia.org/wiki/Reverse_engineering). [Geopend 07 05 2014].
- [7] M. Hamburg, „Analyserapport,” 2014.
- [8] Blackmagic, „Blackmagic driver,” Blackmagic, [Online]. Available: <http://www.blackmagicdesign.com/support/detail/register?sid=3945&pid=4012&fid=0&did=42823&os=linux&alt=true&sdk=false&archived=true>. [Geopend 10 05 2014].
- [9] B. Henderson, „misc: device driver for "miscellaneous" character devices,” About.com, [Online]. Available: [http://linux.about.com/od/lkm\\_howto/a/hwtlkm16t06.htm](http://linux.about.com/od/lkm_howto/a/hwtlkm16t06.htm). [Geopend 28 05 2014].
- [10] J. Corbet, A. Rubini en G. Kroah-Hartman, „The file structure,” in *Linux Device Drivers*, Sebastopol, O'Reilly Media Inc, 2005, pp. 51-52.
- [11] J. Corbet, A. Rubini en G. Kroah-Hartman, „IOCTL,” in *Linux Device Drivers*, Sebastopol, O'Reilly Media Inc, 2005, p. 136.
- [12] Blackmagic, „Support centre Decklink SDK,” [Online]. Available: <http://www.blackmagicdesign.com/support/detail/register?sid=3945&fid=0&did=59001&alt=true&sdk=true>. [Geopend 04 05 2014].
- [13] Linfo, „Pipes: A Brief Introduction,” Linfo, 23 08 2006. [Online]. Available: <http://www.linfo.org/pipes.html>. [Geopend 30 05 2014].
- [14] wIKIMEDIA, „File:ASCII-Table-wide.svg,” wIKIMEDIA, 24 05 2014. [Online]. Available: <http://commons.wikimedia.org/wiki/File:ASCII-Table-wide.svg>. [Geopend 28 05 2014].
- [15] „Objdump,” 23 04 2013. [Online]. Available: <http://unixhelp.ed.ac.uk/CGI/man-cgi?objdump+1>. [Geopend 08 05 2014].
- [16] Wikipedia, „Disassembler,” Wikipedia, 10 10 2013. [Online]. Available: <http://nl.wikipedia.org/wiki/Disassembler>. [Geopend 08 05 2014].

- [17] D. Swanepoel, „Linux Assembly Tutorial,” 19 04 2002. [Online]. Available: [http://docs.cs.up.ac.za/programming/asm/derick\\_tut/](http://docs.cs.up.ac.za/programming/asm/derick_tut/). [Geopend 13 05 2014].
- [18] File systems and Storage Lab, „access\_ok User Space Memory Access,” File systems and Storage Lab, [Online]. Available: <http://www.fsl.cs.sunysb.edu/kernel-api/re243.html>. [Geopend 24 05 2014].
- [19] Wikipedia, „Name mangling,” Wikipedia, 24 05 2014. [Online]. Available: [http://en.wikipedia.org/wiki/Name\\_mangling](http://en.wikipedia.org/wiki/Name_mangling). [Geopend 03 06 2014].
- [20] J. Gaspar, „[http://www.boost.org/doc/libs/1\\_55\\_0/doc/html/circular\\_buffer.html](http://www.boost.org/doc/libs/1_55_0/doc/html/circular_buffer.html),” Boost, 28 10 2013. [Online]. Available: [http://www.boost.org/doc/libs/1\\_55\\_0/doc/html/circular\\_buffer.html](http://www.boost.org/doc/libs/1_55_0/doc/html/circular_buffer.html). [Geopend 02 05 2014].
- [21] D. Miller , R. Henderson en J. Jelinek , „Using Streaming DMA mappings,” Dynamic DMA mapping, [Online]. Available: <http://lwn.net/2001/0712/a/dma-interface.php3>. [Geopend 30 05 2014].
- [22] J. Corbet, A. Rubini en G. Kroah-Hartman, „Top and Bottom Halves,” in *Linux device drivers*, Sebastopol, O'Reilly, 2005, pp. 275 - 277.
- [23] Wikipedia, „Video4Linux,” Wikipedia, 17 04 2014. [Online]. Available: <http://en.wikipedia.org/wiki/Video4Linux>. [Geopend 07 05 2014].
- [24] Linuxtv, „Video for Linux Two API Specification,” Linuxtv, [Online]. Available: [http://www.linuxtv.org/downloads/legacy/video4linux/API/V4L2\\_API/spec-single/v4l2.html](http://www.linuxtv.org/downloads/legacy/video4linux/API/V4L2_API/spec-single/v4l2.html). [Geopend 07 05 2014].
- [25] Debian, „Downloadpagina voor linux-source-3.2\_3.2.57-3+deb7u1\_all.deb,” Debian, [Online]. Available: <https://packages.debian.org/nl/wheezy/all/linux-source-3.2/download>. [Geopend 28 05 2014].
- [26] Kernel.org, „Overview of the V4L2 driver framework,” Kernel.org, [Online]. Available: <https://www.kernel.org/doc/Documentation/video4linux/v4l2-framework.txt>. [Geopend 04 05 2014].
- [27] P. Kruchten, „Prototypes,” in *The rational unified process*, Pearson education, 2009, pp. 190 - 101.
- [28] M. Hamburg, „Architectuurrapport,” 2014.
- [29] P. Kruchten, „The elaboration phase,” in *The Ration Unified Process*, Pearson education, 2009, p. 71.
- [30] Wikipedia, „Requirement,” Wikipedia, 05 02 2014. [Online]. Available: [http://nl.wikipedia.org/wiki/Requirement#Functionele\\_requirements](http://nl.wikipedia.org/wiki/Requirement#Functionele_requirements). [Geopend 07 05 2014].
- [31] Wikipedia, „H.264,” Wikipedia, 24 01 2014. [Online]. Available: <http://nl.wikipedia.org/wiki/H.264>. [Geopend 11 05 2014].

- [32] Wikipedia, „Flash video,” Wikipedia, 27 03 2014. [Online]. Available: [nl.wikipedia.org/wiki/FLV](http://nl.wikipedia.org/wiki/FLV). [Geopend 11 05 2014].
- [33] Wikipedia, „Real Time Messaging Protocol,” Wikipedia, 02 05 2014. [Online]. Available: [http://en.wikipedia.org/wiki/Real\\_Time\\_Messaging\\_Protocol](http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol). [Geopend 11 05 2014].
- [34] M. Hamburg, „Ontwikkeldrapport,” 2014.
- [35] B. van Ingen Schenau, „Software design for non object oriented paradigm,” Stackexchange, 18 11 2012. [Online]. Available: <http://programmers.stackexchange.com/questions/176283/software-design-for-non-object-oriented-paradigm>. [Geopend 04 06 2014].
- [36] Wikipedia, „Kernel panic,” Wikipedia, 13 04 2014. [Online]. Available: [http://en.wikipedia.org/wiki/Kernel\\_panic](http://en.wikipedia.org/wiki/Kernel_panic). [Geopend 10 05 2014].
- [37] J. Corbet, A. Rubini en G. Kroah-Hartman, Linux device drivers, Stanford: O'Reilly, 2005.
- [38] Kernel.org, „The Linux Kernel API,” Kernel.org, [Online]. Available: <https://www.kernel.org/doc/html/docs/kernel-api/>. [Geopend 04 05 2014].
- [39] Wikipedia, „Scrum (softwareontwikkelmethode),” Wikipedia, 09 04 2014. [Online]. Available: [http://nl.wikipedia.org/wiki/Scrum\\_%28softwareontwikkelmethode%29#Werkwijze](http://nl.wikipedia.org/wiki/Scrum_%28softwareontwikkelmethode%29#Werkwijze). [Geopend 12 04 2014].
- [40] P. Kruchten, The Rational Unified Process An Introduction, Upper Saddle River: Pearson Education Inc, 2009.
- [41] R. Grit, Project management, Houten: Noordhoff Uitgevers Groningen, 2008.
- [42] Wikipedia, „PRINCE2,” Wikipedia, 22 03 2014. [Online]. Available: <http://nl.wikipedia.org/wiki/PRINCE2>. [Geopend 12 04 2014].
- [43] makelinux, „Debugging System Faults,” makelinux, [Online]. Available: <http://www.makelinux.net/ldd3/chp-4-sect-5>. [Geopend 30 04 2014].
- [44] Wikipedia, „PRINCE2,” Wikipedia, 22 03 2014. [Online]. Available: <http://nl.wikipedia.org/wiki/PRINCE2>. [Geopend 01 05 2014].
- [45] Wikipedia, „Bus mastering,” Wikipedia, 21 11 2013. [Online]. Available: [http://en.wikipedia.org/wiki/Bus\\_mastering](http://en.wikipedia.org/wiki/Bus_mastering). [Geopend 02 05 2014].
- [46] W. Young, „What is the difference between kernel drivers and kernel modules?,” stackexchange, 05 11 2012. [Online]. Available: <http://unix.stackexchange.com/questions/47208/what-is-the-difference-between-kernel-drivers-and-kernel-modules>. [Geopend 07 05 2014].
- [47] Wikibooks, „x86 Disassembly/The Stack,” Wikibooks, 30 03 2014. [Online]. Available: [http://en.wikibooks.org/wiki/X86\\_Disassembly/The\\_Stack](http://en.wikibooks.org/wiki/X86_Disassembly/The_Stack). [Geopend 07 05 2014].

- [48] M. Hamburg, „Risiko's,” in *Vision document voor het project “Ontwikkelen Video4Linux2 driver voor BlackMagic video capturekaart”*, Waddinxveen, 2014, pp. 17 - 20.
- [49] P. Kruchten, „The concept of Risk,” in *The rational unified Process*, Pearson education, 2009, pp. 118 - 121.
- [50] P. Kruchten, „The inception Phase,” in *The Rational Unified Proces*, Pearson education, 2009, p. 68.
- [51] M. Hamburg, „Mapping blackmagic I/O controls,” in *Architectuur analyse voor het project “Ontwikkelen Linux driver voor Blackmagic video capturekaart”*, Waddinxveen, 2014, pp. 21 - 24.
- [52] J. Warmer en A. Kleppe, „use-case,” in *Praktisch UML*, Pearson Education Benelux, 2008, p. 89.
- [53] J. Warmer en A. Kleppe, *Praktisch UML*, Amsterdam: Pearson Education Benelus, 2008.

## Gebruikte afbeeldingen en tabellen

TABEL 1 FUNCTIONELE KANDIDAAT SYSTEEMEISEN .....	10
TABEL 2 USE-CASE BESCHRIJVING DRIVER LADEN .....	11
TABEL 3 USE-CASE BESCHRIJVING DRIVER VERWIJDEREN .....	11
TABEL 4 IMPACT SCORE .....	12
TABEL 5 KANS SCORE .....	12
TABEL 6 RISICO SCORE .....	12
TABEL 7 DEFECTE HARDWARE.....	12
TABEL 8 SOFTWARE RAAKT VERLOREN .....	13
TABEL 9 PARAMETERS VOOR HET PROGRAMMA CAPTURE.....	18
TABEL 10 DISPLAYMODES.....	19
TABEL 11 ASSEMBLY REGISTERS [17] .....	20
TABEL 12 MANGLED NAME PARAMETERS .....	20
TABEL 13 I/O CONTROL MAPPING.....	21
TABEL 14 /PROC/INTERRUPTS .....	23
TABEL 15 FUNCTIONELE SYSTEEMEISEN .....	28
TABEL 16 NIET FUNCTIONELE SYSTEEMEISEN.....	28
TABEL 17 USE-CASE DRIVER INSTELLEN .....	31
TABEL 18 USE-CASE VIDEO CAPTUREN.....	31
TABEL 19 TOEGEVOEGDE FUNTIES AAN DE BLACKMAGIC_PROBE.....	35
TABEL 20 TEST DRIVER LADEN .....	38
TABEL 21 AUTEUR GEGEVENS.....	48
TABEL 22 GEGEVENS CSNET.....	48
TABEL 23 EXAMINATOREN.....	48
TABEL 24 VERSIEBEHEER .....	48
FIGUUR 1 ORGANIGRAM ORGANISATIE CS NET.....	6
FIGUUR 2 OVERZICHT VAN DE HUIDIGE INFRASTRUCTUUR .....	8
FIGUUR 3 EERSTE USE-CASE-DIAGRAM .....	10

FIGUUR 4 FLOW VAN DE HUIDIGE BLACKMAGIC DRIVER .....	15
FIGUUR 5 HUIDIGE WERKING VAN DE DRIVER.....	17
FIGUUR 6 CIRCULAIR GEHEUGEN [20] .....	22
FIGUUR 7 GLOBALE NETWERK INFRASTRUCTUUR .....	25
FIGUUR 8 HUIDIGE WERKING VAN DE DRIVER.....	26
FIGUUR 9 STRUCTUUR HUIDIGE DRIVER .....	27
FIGUUR 10 NIEUWE SITUATIE DRIVER.....	29
FIGUUR 11 TOTALE USE-CASE.....	30
FIGUUR 12 SEQUENTIE DIAGRAM DRIVER LADEN.....	32
FIGUUR 13 DRIVER VERWIJDEREN .....	33
FIGUUR 14 TOESTAND DIAGRAM DRIVER .....	33
VOORBEELD 1 LOSSE FUNCTIE VOOR DEBUG DOELEINDEN.....	18
VOORBEELD 2 SYSTEEMLOG VAN DE INITIALISATIE EN AANROEP DOESSUPPORTDISPLAYMODE .....	19
VOORBEELD 3 KERNEL PANIC.....	36



## Bijlage 1: Projectgegevens

### Auteur gegevens

Naam	Email adres	Telefoonnummer	studentnummer
<b>Maarten Hamburg</b>	<a href="mailto:Maarten.hamburg@gmail.com">Maarten.hamburg@gmail.com</a>	0626871380	20064311

Tabel 21 Auteur gegevens

### Gegevens CSNet

Naam	Email adres	Telefoonnummer	Functie
<b>Andre Canrinus</b>	<a href="mailto:acanrinus@corp.csnet.nl">acanrinus@corp.csnet.nl</a>	0182-750571	Manager afdeling ontwikkeling
<b>Daan van Heteren</b>	<a href="mailto:d.vanheteren@corp.csnet.nl">d.vanheteren@corp.csnet.nl</a>	0182-570500	Directeur

Tabel 22 Gegevens CSNet

### Examinatoren

Naam	Email adres	Telefoonnummer	Functie
<b>J.J. Visser</b>	<a href="mailto:J.J.Visser@hhs.nl">J.J.Visser@hhs.nl</a>	015-2606281	Begeleidend examinator
<b>J.P.M de Vreught</b>	<a href="mailto:J.P.M.deVreught@hhs.nl">J.P.M.deVreught@hhs.nl</a>	070- 4458521	2 <sup>e</sup> examinator

Tabel 23 Examinatoren

### Versiebeheer plan van aanpak

Datum	Versie	Status
<b>17-12-2013</b>	0.1	Concept
<b>24-02-2014</b>	0.2	Concept
<b>13-03-2014</b>	0.3	Concept
<b>31-03-2014</b>	0.4	Concept
<b>08-05-2014</b>	0.5	Concept
<b>13-05-2014</b>	0.7	Concept
<b>13-05-2014</b>	0.8	Concept
<b>04-06-2014</b>	1.0	Definitief

Tabel 24 Versiebeheer

Bijlage 2: Afstudeerplan

## Afstudeerplan

### Informatie afstudeerder en gastbedrijf (*structuur niet wijzigen*)

**Afstudeerblok:** 2014-1.1 (start uiterlijk 10 februari 2014)

**Startdatum uitvoering afstudeeropdracht:** 02-12-2013

**Inleverdatum afstudeerdossier volgens jaarrooster:** 06-06-2013

**Studentnummer:** 20064311

**Achternaam:** dhr Hamburg

**Voorletters:** M.W.M

**Roepnaam:** Maarten

**Adres:** Willem de Rijkelaan 64

**Postcode:** 2741VN

**Woonplaats:** Waddinxveen

**Telefoonnummer:**

**Mobiel nummer:** 0626871380

**Privé emailadres:** maarten.hamburg@gmail.com

**Opleiding:** Technische Informatica

**Locatie:** Den Haag

**Variant:** deeltijd c.q. avond

**Naam studieloopbaanbegeleider:** T. Androli

**Naam begeleidend examiner:** J. Visser

**Naam tweede examiner:** H. de Vreught

**Naam bedrijf:** CS Engineering

**Afdeling bedrijf:** Ontwikkeling

**Bezoekadres bedrijf:** Brugweg 56

**Postcode bezoekadres:** 2741KZ

**Postbusnummer:**

**Postcode postbusnummer:**

**Plaats:** Waddinxveen

**Telefoon bedrijf:** 0182750500

**Telefax bedrijf:**

**Internetsite bedrijf:** www.csnet.nl

**Achternaam opdrachtgever:** Dhr van Heteren

**Voorletters opdrachtgever:** D

**Titulatuur opdrachtgever:** ING

**Functie opdrachtgever:** Directeur

**Doorkiesnummer opdrachtgever:** 0182750500

**Email opdrachtgever:** d.vanheteren@corp.csnet.nl

**Achternaam bedrijfsmentor:** dhr Canrinus

**Voorletters bedrijfsmentor:** A

**Titulatuur bedrijfsmentor:** ING

**Functie bedrijfsmentor:** Manager afdeling ontwikkeling

**Doorkiesnummer bedrijfsmentor:** 0182-750571

**Email bedrijfsmentor:** [acanrinus@corp.csnet.nl](mailto:acanrinus@corp.csnet.nl)

*NB: bedrijfsmentor mag dezelfde zijn als de*

*opdrachtgever*

**Doorkiesnummer afstudeerder:** 0182750574

**Functie afstudeerder (deeltijd/duaal):** Deeltijd

**Titel afstudeeropdracht:**

## Ontwikkelen Linux driver voor BlackMagic video capturekaart

### Opdrachtomschrijving

#### 1. Bedrijf

CS Engineering is een handelsnaam van Communications Security Net B.V. Vanuit CS Engineering wordt in-house software ontwikkeld voor intern gebruik en voor klanten van Communications Security Net B.V.

Communications Security Net B.V is een bedrijf welke behoort tot het MKB (Midden Klein Bedrijf) met 19 personeelsleden.

DCT Net is een dochter onderneming van Communications Security Net B.V. DCT Net is operationeel partner van de sIKN (stichting Intermediair Kerkomroep Nederland). Vanuit DCT Net wordt hard- en software geleverd ten behoeve van streaming diensten voor kerken. Voor deze streaming diensten levert DCT Net zendapparatuur alsmede apparatuur waarmee de stream kan worden beluisterd.

Omdat CS Engineering haar klanten wil voorzien van maatwerk, wordt er binnen Cs Engineering gebruik gemaakt van diverse programmeer talen om zodoende te voldoen aan de wensen van de klant.

Enkele producten die vanuit CS Engineering worden ontwikkeld zijn:

**HDN:** Dit is een product dat wordt gebruikt in de financiële markt voor het versturen en ontvangen van hypotheek en opvragen van taxatie rapporten(<http://www.hdn.nl>).

**Setrax:** Is een product voor het opzetten van een tunnelverbinding tussen een host station en een doel station. Om deze verbinding op te zetten is een hardware sleutel benodigd evenals een gebruikersnaam en wachtwoord(<http://www.setrax.nl>).

**sIKN:** Is een stichting waarvoor CS Engineering software maakt. Het gaat hierbij dan om software t.b.v. Streaming diensten(<http://www.sikn.nl> <http://www.kerkomroep.nl>).

DCT Net heeft voor het bekijken van video diensten een nieuwe zender aangeschaft. Deze zender maakt gebruik van een BlackMagic HD capturekaart. Voor het streamen van video wordt binnen de organisatie FFMPEG gebruikt. FFMPEG is een cross-platform applicatie voor het opnemen, converteren en streamen van audio en video. Daarnaast wordt video gebruikt in combinatie met Video4Linux2. Video4Linux2 is een video capture API voor Linux. Hiermee kan video van diverse recorders worden gecaptured.

#### 2. Probleemstelling

De huidige driver van BlackMagic geeft als video data, onbewerkte video data, echter kan FFMPEG niet met deze onbewerkte video data overweg. Hiervoor is momenteel een eigen implementatie ontwikkeld waardoor de raw video data wordt omgezet naar video4linux2 ten behoeve van FFMPEG, echter wordt met deze implementatie de video data 2x meer verwerkt dan wenselijk is. Dit zorgt voor een ongewenste hogere processor load en geheugen verbruik.

#### 3. Doelstelling van de afstudeeropdracht

Het doel van de opdracht is om met een driver voor de BlackMagic capturekaart de onbewerkte video data direct om te zetten naar video4linux2 met verminderd gebruik van de resources.

#### 4. Resultaat

Het resultaat is een driver voor de BlackMagic HD capturekaart.

#### 5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

- Oriëntatie fase 5 dagen
  - Verhelderen opdracht
  - Planning opstellen
  - Risicoanalyse
- Analysefase 25 dagen
  - Bestuderen literatuur
  - Reverse engineering BlackMagic capturekaart.
  - Achterhalen hoe de capturekaart werkt.
  - Opstellen systeemeisen.
- Ontwerpfase 10 dagen
  - Architectuur ontwerpen voor de driver
- Ontwikkelfase 45 dagen
  - Basis driver
    - Ontwerp uitwerken
    - Realiseren
    - Testen
  - Full HD driver
    - Ontwerp uitwerken
    - Realiseren
    - Testen
  - Acceptatietest afnemen

#### 6. Op te leveren (tussen)producten

- *Plan van Aanpak*
- *Analyserapport*
- *Architectuurrapport*
- *Ontwikkeldrapport*
  - *Ontwerp driver*
  - *Testrapportage*

#### 7. Te demonstreren competenties en wijze waarop

Voor het bepalen van de complexiteit wordt onderstaand schema gebruikt:

Competentiecode	niveau	Competentietitel	Wijze waarop
G1	3	Praktische aspecten hanteren in (internationale) projecten	Risicoanalyse in het plan van aanpak
A1	4	Analyseren van het probleemdomen	Beschrijven Reverse engineering & achterhalen werking capturekaart in analyserapport
C10	3	Ontwerpen van een systeemarchitectuur	Architectuurrapport
C8	3	Ontwerpen van een technisch informatie systeem	De ontwerpen in het ontwikkelrapport
D17	3	Testen van software systemen	Testrapportages bij het ontwikkelrapport