



# Installatie van FFmpeg voor RTSP te converteren naar SRT

2023

JONATHAN DE WIT

## Index

1	Introductie .....	4
2	Installatie van FFmpeg .....	4
2.1	Vorbereiding op de installatie .....	4
2.2	Installatie van de FFmpeg dependances .....	5
2.2.1	SRT van Haivision .....	5
2.2.2	fdk-aac .....	6
2.3	Installeren en configureren van FFmpeg.....	7
3	Converteren van RTSP naar SRT .....	9
4	Literatuurlijst .....	10

# 1 Introductie

Voor het RTSP-video signaal van mijn ESP32-CAM bord te converteren naar een SRT signaal zal ik FFmpeg gebruiken. FFmpeg is een open source command line tool en framework voor het opnemen, converteren en streamen van audio en video.

FFmpeg gebruikte ik op een Raspberry PI 4 B op Raspbian OS, specifiek op de Debian versie Bullseye.

FFmpeg staat niet toe om ons RTSP-signaal naar SRT te converteren met een standaardinstallatie ("sudo apt install ffmpeg"). Dit komt omdat de standaard installatie niet met de benodigde configuratie komt voor de conversie en omdat hij bepaalde essentiële componenten mist om met het SRT-protocol te werken, zoals SRT-tools.

Hierdoor was het nodig om een custom installatie te maken van FFmpeg om de conversie te doen werken.

## 2 Installatie van FFmpeg

Zoals eerder vermeld is het noodzakelijk om een custom installatie van FFmpeg uit te voeren.

Dit houdt het volgende in:

1. Installeren van de vereiste pakketten die nodig zijn voor FFmpeg te installeren.
2. Installeren van alle dependenties voor te werken met FFmpeg.
3. Configureren en installeren van FFmpeg.

### 2.1 Voorbereiding op de installatie

Voor dat men beginnen aan de installatie van FFmpeg is het belangrijker dat VLC niet geïnstalleerd is omdat dit voor conflicten kan zorgen tijdens de installatie.

VLC en zijn dependenties kan men gemakkelijk verwijderen met behulp van de volgende commando's:

```
//Verwijderen van alle VLC packages  
sudo apt purge vlc*  
  
//Verwijderen van alle dependances die men niet meer nodig hebben  
sudo apt autoremove
```

Jonathan De Wit

Voor FFmpeg en zijn dependencies te kunnen installeren, is het noodzakelijk om de volgende pakketten te installeren. Deze zullen nodig zijn voor FFmpeg en zijn componenten te configureren, compileren en te installeren.

Deze kunnen na de installatie zonder probleem verwijderd worden.

```
//Best practice om eerst updaten  
sudo apt update && sudo apt upgrade  
  
//Installeren de nodige packages voor de installatie  
sudo apt install autoconf dh-autoreconf automake build-essential cmake pkg-  
config texinfo wget git yasm nasm tcl tclsh libtool libtheora-dev libva-dev  
libgpg-error-dev libass-dev libfreetype6-dev libgnutls28-dev libsdl2-dev libsdl1.2-  
dev libvdpau-dev libvorbis-dev libxcb1-dev libxext-dev libx11-dev libxfixes-dev  
libxcb-shm0-dev libxcb-xfixes0-dev zlib1g-dev libssl-dev libx264-dev libx265-  
dev libnuma-dev libx265-doc libvpx-dev libmp3lame-dev libopus-dev texi2html  
zlib1g-dev libopus-dev
```

## 2.2 Installatie van de FFmpeg dependances

### 2.2.1 SRT van Haivision

De belangrijkste dependency die geïnstalleerd moet worden is SRT van Haivision (oprichters van de SRT-alliantie). Deze dependency maakt het mogelijk om SRT-live-transmit te gebruiken om live video te verzenden met behulp van het SRT protocol.

De installatie gaat als volgt:

```
//Maken van een folder voor waar men de dependency gaan downloaden  
mkdir Documents/ffmpegInstall && cd Documents/ffmpegInstall  
  
//Downloaden van de SRT-git repository van Haivision  
git clone https://github.com/Haivision/srt.git  
  
//Veranderen van de huidige werkmap naar srt  
cd srt  
  
//Dit zal de configure script van de srt package uitvoeren.  
//Dit script zal nagaan als het systeem alle nodige dependencies bezit en zal  
//een Makefile aanmaken voor het pakket te bouwen.  
  
../configure  
  
//Dit commando zal de Makefile uitvoeren voor het pakket te bouwen.  
make  
  
//Installeren van de package op het systeem.  
sudo make install
```

### 2.2.2 fdk-aac

Deze package biedt een AAC-encoder die gebruikt kan worden als men ook audio willen doorsturen. Ik heb dit voor mijn project niet nodig omdat ik geen microfoon heb op mijn ESP32-CAM. Maar het is well interessant om hem te installeren voor future proof te zijn in het geval men ook later audio willen doorsturen.

De installatie van fdk-aac gaat als volgt:

```
//Dit commando zal het fdk-aac repository downloaden.
git clone --depth 1 https://github.com/mstorsjo/fdk-aac

//Hier zal men $HOME/bin plaatsen in de PATH environment variable.
//$HOME/bin is een folder die de geïnstalleerde binaries van een gebruiker
//worden geplaatst.
PATH="$HOME/bin:$PATH"

//Hier zal men in een andere environment variable "PKG_CONFIG_PATH" het
//path "/usr/local/lib/pkgconfig" plaatsen. Dit path is de locatie waar de
//package config files opgeslagen zijn.
export PKG_CONFIG_PATH="/usr/local/lib/pkgconfig"

//Veranderen van de huidige werkdirectory naar fdk-aac
cd fdk-aac

//Hier zal men de autoreconf tool gebruiken om van de broncode de nodige files
//te genereren om de package te kunnen bouwen. De -fiv parameter zal de files
//overschrijven als deze al bestaan. De parameter zorgt er ook voor dat de
//output weergegeven wordt.
autoreconf -fiv

//Dit zal de configure script van de fdk-aac package uitvoeren.
//Dit script zal nagaan als het systeem alle nodige dependencies bezit en zal
//een Makefile aanmaken voor het pakket te bouwen.
./configure

//Dit commando zal de Makefile uitvoeren voor het pakket te bouwen
make

//Installeren van de package op het systeem.
sudo make install
```

## 2.3 Installeren en configureren van FFmpeg

Nu dat alle dependencies geïnstalleerd zijn zal men FFmpeg kunnen installeren en configureren.

De installatie en configuratie van FFmpeg gaat als volgt:

```
//Dit commando zal het FFmpeg repository downloaden.
git clone https://github.com/FFmpeg/FFmpeg.git

//Hier zal men $HOME/bin plaatsen in de PATH environment variable.
//$HOME/bin:$PATH is een folder die de geïnstalleerde binaries van een
gebruiker //worden geplaatst.
PATH="$HOME/bin:$PATH"

//Hier zal men in een andere environment variable "PKG_CONFIG_PATH" het
//path "/usr/local/lib/pkgconfig" plaatsen. Dit path is de locatie waar de
//package config files opgeslagen zijn.
export PKG_CONFIG_PATH="/usr/local/lib/pkgconfig"

//Veranderen van de huidige werkmapper naar FFmpeg
cd FFmpeg

//Dit zal de configure script van de FFmpeg package uitvoeren.
//Dit script zal nagaan als het systeem alle nodige dependencies bezit en zal
//een Makefile aanmaken voor het pakket te bouwen.
//Ik zal hier ook een groot aantal parameters aan toevoegen die nodig zijn
//voor een RTSP signaal te converteren naar een SRT signaal
sudo ./configure --extra-ldflags="-latomic" --arch=armel --target-os=linux --
enable-gpl --enable-libmp3lame --enable-libfdk-aac --enable-libfreetype --
enable-libx264 --enable-libx265 --enable-vaapi --enable-nonfree --enable-
mmal --enable-libsrt

//Dit commando zal de Makefile uitvoeren voor het pakket te bouwen
sudo make

//Installeren van de package op het systeem.
sudo make install

source ~/.profile

//Na de installatie is het nodig om het systeem opnieuw op te starten
sudo reboot

sudo ldconfig
```

Jonathan De Wit

Zoals U het misschien hebt opgemerkt, heb ik tijdens de installatie van FFmpeg veel parameters mee aan mijn ./configure. Hieronder vind u een uitleg van de belangrijkste parameters die ik meegeef:

- '--target-os=linux': deze flag specificeert dat FFmpeg geconfigureerd moet zijn voor een Linux operating system.
- '--enable-libfdk-aac': deze flag maakt het mogelijk om de dependance 'fdk-aac' op te nemen die ik eerder heb geïnstalleerd om audio in AAC te ondersteunen.
- '--enable-libx264 --enable-libx265': Deze 2 flags voegt support toe aan FFmpeg voor H.264 en H.265 encoding door gebruik te maken van de x264 en x265 codex.
- '--enable-vaapi': Deze 2 flags voegt support voor hardware acceleration voor een snellere encoding van video door middel van OpenMAX.
- '--enable-libsrt': Deze flag maakt het mogelijk om het SRT protocol te gebruiken aan de hand van de dependency die ik eerder heb geïnstalleerd.

## 3 Converteren van RTSP naar SRT

Nu dat FFmpeg geïnstalleerd is, kan ik het gebruiken om mijn RTSP live video stream van mijn ESP32-CAM (encoded in MJPEG) te converteren naar een SRT live videostream (encoded in H.264 met een pixelformat YUV420p) die doorgestuurd word naar mijn Nimble media server

Voor deze conversie uit te voeren gebruik ik het volgende commando:

```
ffmpeg -i rtsp://192.168.1.28:8554/mjpeg/1 -pix_fmt yuv420p -c:v libx264 -r 30 -b:v 1000k -f mpegts srt://192.168.1.32:40005
```

De parameters van dit commando doen het volgende:

- '-i rtsp://192.168.1.28:8554/mjpeg/1': Deze parameter zal het RTSP signaal van mijn ESP32-CAM nemen als inputstream
- '-pix\_fmt yuv420p': Deze parameter zal de originele pixel formaat van de ESP32-CAM converteren naar yuv420p. Dit doe ik omdat het originele Formaat niet kan gelezen worden door de Android telefoons waarmee ik het getest heb.
  - o Let op dat zonder deze conversie het formaat nog altijd leesbaar gaat zijn in een Android VM in Android studio of in VLC maar niet op een fysieke Android telefoon.
- '-c:v libx264': Deze parameter zal zorgen dat de inkomende MJPEG video stream word geconverteerd naar een ander video codec in dit geval H.264. Dit is nodig omdat MJPEG niet word ondersteund in Nimble server met het SRT protocol.
- '-r 30': Dit dient om een frame rate te hebben van 30 frames per seconde.
- '-b:v 1000k': Dit plaats de bitrate van de output video aan 1000 kilobits per seconde.
- '-f mpegts' Deze parameter specificeert dat het output formaat MPEG-TS digitale container formaat gebruikt.
- 'srt://192.168.1.32:40005': Dit specificeert het SRT destinatie adres.



## 4 Literatuurlijst

FFmpeg doc. (S.D.). Compile FFMpeg for Ubuntu, Debian, or Mint. FFMpeg. Op <https://trac.ffmpeg.org/wiki/CompilationGuide/Ubuntu>

FFmpeg (S.D.). FFMpeg Documentation. FFMpeg. Op <https://ffmpeg.org/ffmpeg.html>

Shara Bayko, M. (2023, 13 februari). Secure reliable transport (SRT) Protocol. Github. Op <https://github.com/Haivision/srt>

Haivision. (S.D.). About us. Haivision. Op <https://www.haivision.com/about/#0>

Richter, T. (2022, 8 juny). Converting an RTSP stream to SRT on a RaspberryPi. Github. Op <https://github.com/richtertoralf/rtsp2srt>