

Masterproef FTI: Elektronica-ICT Voortgangverslag



Voornaam, Naam	Dennis Joosens	E-mail:	dennis.joosens@student.uantwerpen.be	VGV	5
-----------------------	-----------------------	----------------	---	------------	----------

ACADEMIEJAAR 2016/2017

VERSLAG INGEDIEND OP: 23/04/2017

Voornaam, naam promotor(s)				
<p>Theo Debrouwere</p> <p>+32 470 653 615</p> <p>t.debrouwere@televic.com</p> <p>Walter Daems</p> <p>+32 473 335 155</p> <p>walter.daems@uantwerpen.be</p>				
Data waarop de rapporten werden ingediend	1. 26/02/2017	2. 12/03/2017	3. 26/03/2017	4. 09/04/2017
	5. 23/04/2017	6.	7.	8.
	9.	10.	11.	12.

ABSTRACT VAN HET ONDERZOEK

Ontwerpen van een proof of concept videoconferencing systeem met een maximale end-to-end latency van 25 ms gebruik makende van het TI AM5728 EVM ontwikkelingsbord met camera module.

Korte omschrijving van de evolutie van het onderzoek tijdens de betrokken periode, met aanduiding van de reeds bekomen resultaten en een planning voor de verdere uitwerking, welke problemen zijn ondervonden en hun oplossingen (totaal minimum twee pagina's - maximum vijf pagina's):

Week 10/04/2017 – 23/04/2017

De opdracht voor deze twee weken bestond uit het aanpassen van het latency.c bestand zodat we een zelf opgewekt impuls signaal kunnen insturen en deze via software kunnen uitsturen via het playback apparaat. Bijkomend de parameters instellen zodat we een minimale latency kunnen bekomen en het audio gedeelte kunnen afsluiten. Bijkomend was het de bedoeling om al te starten met de audio encapsulatie, encoding en opmeten van de netwerk latency.

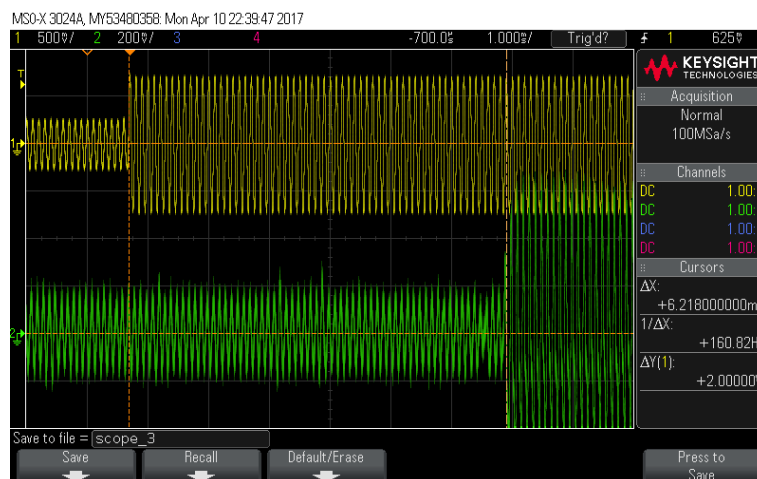
1. Latency opmeten via aanpassing latency.c bestand

De meting werd uitgevoerd door aan de ingang een 10 kHz sinus te leggen die op een bepaald tijdstip van amplitude verandert. Hiermee kunnen we

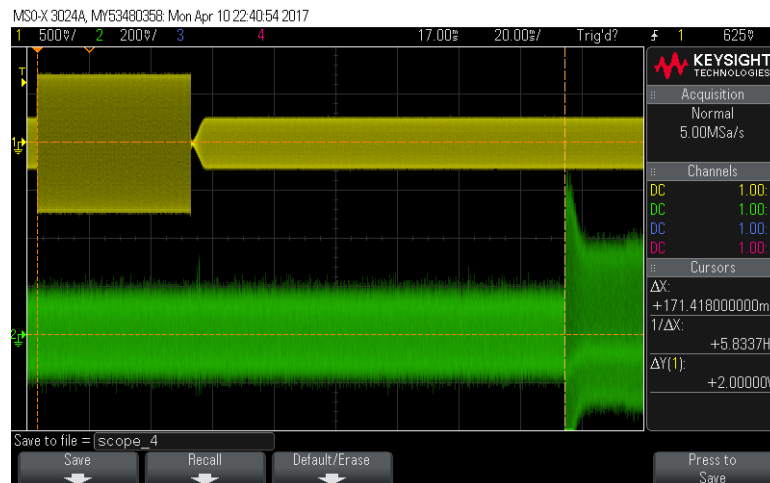
Om het programma te runnen kan je eventueel een bepaald aantal parameters meegeven zoals buffersize, minimum en maximum latency uitgedrukt in frames, samplerate,... . Voor de uitvoering gebruiken we twee kanalen omdat één kanaal niet wordt ondersteund door het EVM board. Verder een samplerate van 48000 en de samples zijn 16 bits Little Endian format. Ook geven we de polling parameter mee om de CPU te ontlasten. Door met de minimum latency parameter te spelen komen we tot de resultaten die je terugvindt in tabel 1.

Minimum latency (in frames)	Opgemeten latency (in ms)
256	6,2
1024	22,2
4096	86,02
8192	171,42

(Tabel 1. Opmeting latency)



(Fig 1. Opmeting laagst verkregen latency)



(Fig 2. Opmeting hoogst verkregen latency)

Enkele opmerkingen hierbij zijn dat als we de polling parameter gebruiken of niet, dit geen effect heeft op de latency. Ook zijn er tijdens het afspelen van spraak of muziek laag frequente spikes te horen. Alsook als we de polling parameter gebruiken is er een high pitch geluid te horen wat best storend is. De reden waar dit vandaan komt is nog onduidelijk. Bijkomend verschilt de gemeten waarde met de gemeten waarde in de software rond +/- 1 ms.

Vb. 1:

$$latency_{sw} = \frac{256 \text{ frames}}{48 \text{ kHz}} = 5,33 \text{ ms}$$

Op de scoop lezen we 6,2 ms.

Vb. 2:

$$latency_{sw} = \frac{8192 \text{ frames}}{48 \text{ kHz}} = 170,67 \text{ ms}$$

Op de scoop lezen we 171,42 ms.

Dit heeft vermoed ik te maken doordat het capture device en playback device tegelijkertijd worden opgestart (linking). Het playback device moet iets afspelen maar moet wachten tot de buffer van het capture device vol is. Als oplossing wordt er in de software twee keer een stilte geschreven naar de buffers. Het wegschrijven van deze buffers heeft een bepaalde tijd nodig. Wat de verhoging in gemeten latency kan verklaren. Een ander aandeel in dit verschil kan de meetfout zijn (plaatsen van de cursors) bij het opmeten met de oscilloscoop.

2. Audio encapsulatie en encoding

Om de audio data te verzenden over een netwerk moeten we de data encapsuleren. Hiervoor willen we Real-time Transport Protocol gebruiken wat gebaseerd is op UDP. RTP is het aangewezen protocol voor audio en video streams en lijkt hierdoor een goede keuze.

Na het encapsuleren gaan we de data proberen te encoderen/decoderen. Het EVM board ondersteunt

standaard AAC en MPEG-2 audio encoding/decoding. We gaan echter afwijken van deze keuze omdat beide een te grote latency veroorzaken. Hierdoor valt onze keuze op Opus. Opus is een verlieslatende audio codec waarmee je een zeer lage latency kan verkrijgen en over het algemeen een lagere bit rate heeft. Deze blijkt zeer geschikt voor real-time applicaties.

Ik ben nu bezig met het implementeren van het encapsulatie gedeelte. Dit gedeelte is nog niet af dus leek me dit niet relevant om hier al te plaatsen.

Extra informatie

Bijgewoonde seminars, presentaties, workshops, bedrijfsbezoeken etc in deze periode (onderwerp, datum, korte samenvatting en beoordeling)

1.

2.

3.

4.

Nieuwe contacten gemaakt in deze periode (naam, voornaam, e-mail, telefoonnummer, bedrijf, functie, extra opmerkingen)

1.

2.

3.

4.

Gelezen artikels, boeken, interviews, etc (titel, auteurs, aantal blzn., korte beschrijving, eigen beoordeling (wat is de meerwaarde voor het onderzoek))

http://www.alsa-project.org/main/index.php/Test_latency.c

http://www.alsa-project.org/alsa-doc/alsa-lib/_2test_2latency_8c-example.html

Boek - CCNA Exploration Companion Guide Network Fundamentals - Mark A. Dye, Rick McDonald, Antoon W. Ruff

<https://tools.ietf.org/html/rfc3550>

<https://tools.ietf.org/html/rfc7587>

Visie en eventuele commentaar van de promotor