

USB-S/PDIF-interface

digitale audio-uitgang voor computer, laptop, tablet of smartphone

Stephan Lück (Duitsland)

De meeste PC's, laptops, tablets en smartphones bezitten geen 'echte' audio-uitgang; meestal zijn alleen een multifunctionele USB-uitgang en een hoofdtelefoon-uitgang voorhanden. Dat levert problemen op wanneer u een van die apparaten op een hoogwaardige versterker of een AV-receiver wilt aansluiten. Om dit probleem voor eens en voor altijd uit de wereld te helpen, presenteren we hier een hoogwaardige USB-S/PDIF-interface.



PROJECT-INFO

Tags

digitale audio, PC, laptop, tablet, smartphone

Moeilijkheid

beginners – gevorderden – experts

Tijd

ca. 4 uur

Gereedschap

soldeergereedschap (SMD en through-hole), mechanisch gereedschap

Kosten

ca. € 35...40

EIGENSCHAPPEN

- › Voedingsspanning +5 V
- › Stroomopname 36...43 mA
- › Microcontroller PIC32MX27F256B-I
- › Ondersteunt
 - 3 bemonsteringsfrequenties: 44,1, 48 en 96 kHz
- › Ondersteunt 16-, 20- en 24-bit audio
- › Optische en elektrische S/PDIF-uitgang
- › IR-afstandsbediening (met de huidige firmware RC5)
- › Werkt met Windows 7/10, Linux, Android en Raspbian

S/PDIF (of S/P-DIF) staat voor "Sony/Philips Digital Interface Format" en is een interface om audiodata in digitale vorm over niet al te grote afstanden over te dragen tussen modules of systemen zoals thuisbios of hifi-apparatuur. Het is een éénrichtings-interface voor seriële data zonder separaat kloksignaal (de klok wordt uit het signaal zelf teruggewonnen). De achterliggende gedachte is om audiodata zo lang mogelijk in het digitale domein te houden en pas op het laatste moment naar een analoog signaal te converteren.

S/PDIF [1] is gebaseerd op de professionele AES3-standaard; op protocolniveau zijn beide standaarden compatibel, maar de elektrische eigenschappen zijn verschillend. Een typische professionele AES3-interface gebruikt driepolige XLR-stekkers en een afgeschermd symmetrische twisted-pair kabel met een impedantie van 110 Ω. Het signaalniveau bedraagt 3...10 V_{tt} (AES/EBU 2...7 V_{tt}). Minder vaak worden BNC-connectoren met 75Ω-coaxkabel gebruikt. Bij de symmetrische AES3 met XLR-stekkers (afgeschermd twisted pair) kan maximaal een afstand van 1000 m worden overbrugd; bij de coaxversie is dat 100 m.

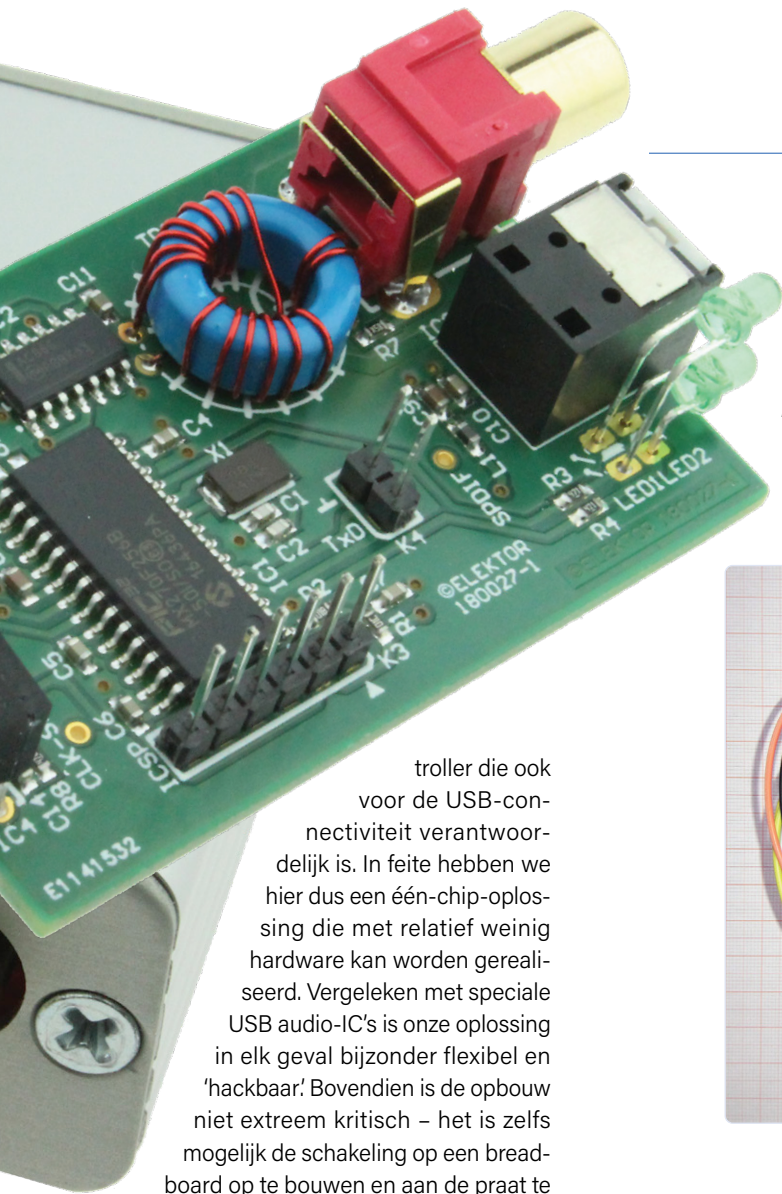
Voor S/PDIF zijn twee soorten verbindingen gedefinieerd. De eenvoudigste variant (die bij de meeste 'betere' audiocomponenten wordt toegepast) bestaat uit een 75Ω-coaxkabel met

(meestal oranje-kleurige) cinch-stekkers. Het signaalniveau bedraagt ongeveer 0,5 V_{tt}. De tweede variant draagt de naam Toslink (van Toshiba Link). Dit is een gestandaardiseerde lichtgeleider met een rode LED (659 nm) als zender. Doorgaand is de lichtgeleider een eenvoudige POF-kabel (Plastic Optical Fibre). Daarmee kunnen afstanden tot zo'n 10 m worden overbrugd. Met hoogwaardige glasvezel is zonder repeater een afstand van maximaal 30 m mogelijk. De optische signalen hebben dezelfde logische opbouw als de elektrische signalen – en dat ligt ook voor de hand omdat het elektrische signaal de LED uitsluitend in- en uitschakelt.

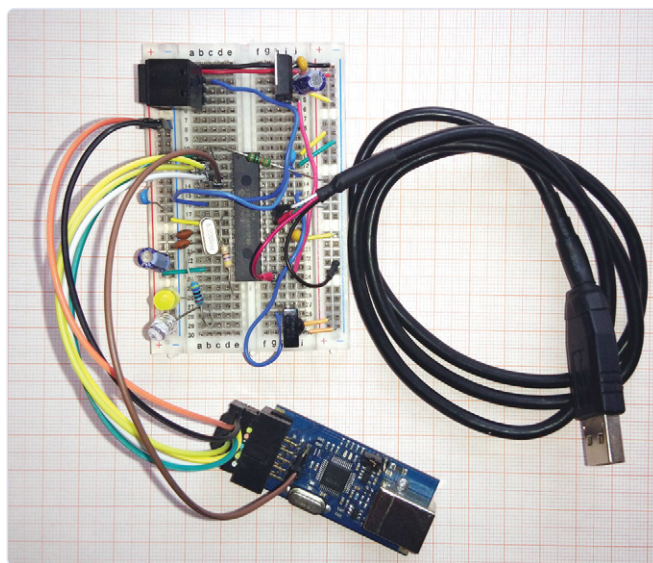
Het project

De schakeling die we hier beschrijven is een USB-naar-S/PDIF omzetter. Via de USB-ingang kan de converter gemakkelijk op PC & Co. worden aangesloten; de S/PDIF-uitgang maakt aansluiting mogelijk op AV-receivers, high-end versterkers of stand-alone audio-DAC's. De S/PDIF-uitgang is dubbel uitgevoerd (elektrisch en optisch); dankzij een IR-ontvanger is afstandsbediening mogelijk. **Figuur 1** geeft een impressie van de opgebouwde schakeling.

De S/PDIF-bitstream wordt geheel softwarematig gegenereerd in dezelfde microcon-



Figuur 1. De USB-S/PDIF-interface is lekker compact.



Figuur 2. Het prototype op breadbord.

troller die ook voor de USB-connectiviteit verantwoordelijk is. In feite hebben we hier dus een één-chip-oplossing die met relatief weinig hardware kan worden gerealiseerd. Vergeleken met speciale USB audio-IC's is onze oplossing in elk geval bijzonder flexibel en 'hackbaar.' Bovendien is de opbouw niet extreem kritisch – het is zelfs mogelijk de schakeling op een breadboard op te bouwen en aan de praat te krijgen (zie **figuur 2**).

Voor dit project hebben we gekozen voor een microcontroller van het type PIC32MX270: deze heeft de periferie voor USB-audio-toepassingen aan boord, en beschikt over voldoende RAM om gecodeerde S/PDIF-frames op te slaan. Daarnaast zijn er van deze controller varianten verkrijgbaar met relatief weinig aansluitpinnen, wat het printontwerp eenvoudiger maakt.

Voor het project is een klein dubbelzijdig printje ontworpen waarop naast de microcontroller ook de voeding en de optische en elektrische S/PDIF-uitgangen een plaatsje hebben gevonden, alsmede een ontvanger voor de IR-afstandsbediening plus twee LED's (deze signaleren de audio-bemonsteringsfrequentie en dat de uitgang actief is). Het printje is zo ontworpen dat het in een klein Hammond-kastje past.

Het audiogedeelte

Het audiodeel van de schakeling omvat een USB-interface conform de USB-audioklasse [2][3], de in software geïmplementeerde S/PDIF-encoder en de S/PDIF-uitgang, waarvoor de SPI-uitgang van de microcontroller wordt gebruikt. Een DMA-kanaal kopieert

de S/PDIF-frames continu van een ringbuffer naar de SPI. In **figuur 3** zijn de software- en hardware-componenten blokschematisch weergegeven.

Omdat de standaard USB-audiospecificatie is geïmplementeerd, hoeven voor de USB-S/PDIF-interface geen speciale drivers op de host te worden geïnstalleerd.

De *device class*-specificatie is best wel complex. Daarom heeft het USB-IF (*USB Implementers Forum*) de *USB Audio Device Class Specification for Basic Audio Devices* gepubliceerd [4], die een kleine subset van de originele *Audio Device Class*-specificatie omvat. De USB audio-interface die in ons project is geïmplementeerd, lijkt sterk op de hoofdtelefoon-applicatie die in [4] wordt beschreven. We hebben daar echter nog een paar extra bemonsteringsfrequenties toegevoegd, de volumeregeling is weggelaten en de USB-descriptor is zodanig aangepast dat die een S/PDIF-uitgang beschrijft in plaats van een luidspreker. Onze USB audio-interface heeft de volgende eigenschappen:

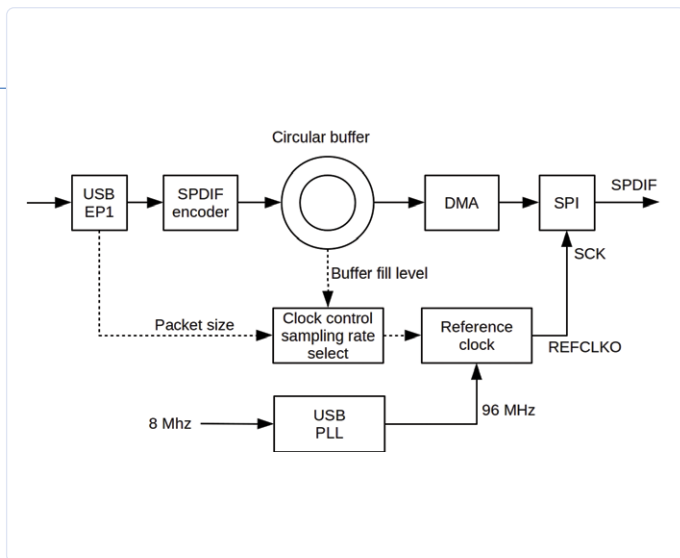
- ondersteuning voor drie bemonsteringsfrequenties (44,1 kHz, 48 kHz en 96 kHz);
- één audioformaat: twee kanalen met 3 bytes (24 bit) per kanaal (S24_3LE);
- een mogelijkheid om het audiosignaal te onderdrukken (mute).

Deze features zijn gespecificeerd in de USB-configuratie-descriptor (dat is het bestand `usb_descriptors.c` in het softwarepakket). Nadere informatie hierover is te vinden in [3], [5] en [6].

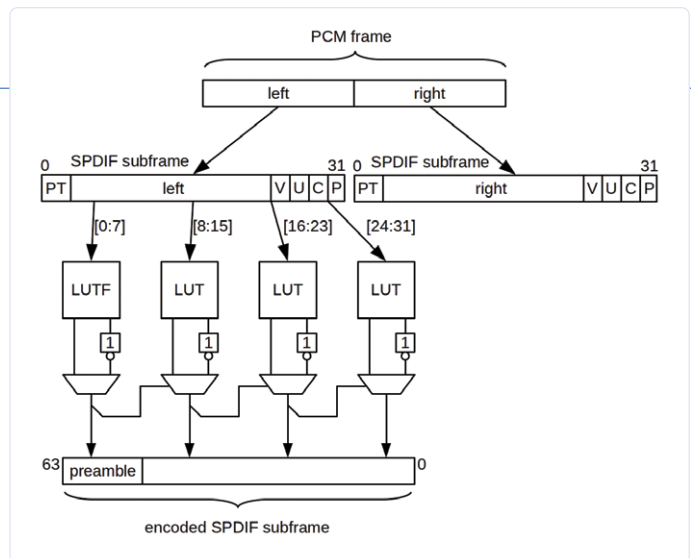
Omdat we bij dit project geen formele USB-compliance nastreven, hebben we vendor- en product-ID's gekozen die gebruikt kunnen worden zonder gevaar van conflicten met officiële USB-producten.

S/PDIF-encoder en seriële uitgang

De USB-interface maakt gebruik van het isochrone USB-eindpunt 1 om audiosamples naar de S/PDIF-interface over te dragen. Dat wil zeggen dat in elk USB-frame van 1 ms één



Figuur 3. De blokschematische opbouw van de interface.



Figuur 4. Principiële opzet van de software-SPDIF-encoder.

USB-datapakket wordt overgedragen. Telkens wanneer de USB-hardware van de microcontroller een nieuw isochroon pakket heeft ontvangen, wordt een interruptroutine aangeroepen die de afzonderlijke PCM-frames uit het USB-datapakket converteert naar de corresponderende S/PDIF-frames. Voor nadere informatie over S/PDIF verwijzen we naar [7] en [8]. In **figuur 4** hebben we blokschematisch geïllustreerd hoe de software-S/PDIF-encoder werkt.

Bovenin de figuur zien we 48 bit grote PCM-frames; elk frame bevat twee samples – een voor het linker kanaal en een voor het rechter kanaal. Deze samples worden geconverteerd in een tussenliggende 32-bit representatie van S/PDIF-subframes door een 4-bit preamble-tag toe te voegen die het type S/PDIF-preamble (X, Y of Z) specificeert dat gebruikt moet worden bij de uiteindelijke codering van het S/PDIF-subframe. Ook worden aan het eind de vier bits V, U, C en P toegevoegd. De V- en U-bits corresponderen met een geldigheidsindicator respectievelijk gebruiker-databits. In onze implementatie zijn deze bits altijd 0. De C-bits bevatten de kanaalstatusdata en de P-bits zijn pariteitsbits voor de afzonderlijke subframes. Deze pariteitsbits worden voor elk subframe berekend met behulp van het geoptimaliseerde algoritme dat in [9] is beschreven. Twee opeenvolgende S/PDIF-subframes vormen een S/PDIF-frame. 192 opeenvolgende S/PDIF-frames vormen samen een S/PDIF-blok. De kanaalstatus wordt verkregen door de C-bits van een blok achter elkaar te zetten – zodoende heeft het kanaalstatus-blok een lengte van 192 bits (24 bytes). Hier gebruiken we de kanaalstatus-bits om de momentele bemonsteringsfrequentie door te

geven aan de ontvanger die op de S/PDIF-uitgang is aangesloten.

S/PDIF gebruikt biphasemark codering (ook bekend onder de naam differentiële Manchester-codering) om de audiodata en de preamble-bits te coderen. Omdat twee codebits met één databit corresponderen, bestaat een gecodeerd S/PDIF-subframe uit 64 bits. Bij de eenvoudigste implementatie van biphasemark codering zou elk bit van een S/PDIF-subframe afzonderlijk gecodeerd moeten worden, wat natuurlijk kostbare processor tijd 'vreet'. Daarom gebruiken we hier twee verschillende opzoektabels (*look-up tables*, LUT's) om telkens één byte tegelijk te coderen. De LUTF-opzoektabel wordt gebruikt om het eerste byte van een S/PDIF-subframe te coderen, terwijl de LUT-opzoektabel voor de resterende drie bytes van het subframe te coderen. De speciale LUTF-tabel bevat de preamble-bitpatronen. In **figuur 4** is opzoektabel LUT driemaal getekend omdat die driemaal wordt gebruikt om één S/PDIF-subframe te coderen. In werkelijkheid staat er maar één instantie van die tabel in het geheugen. De opzoektabels worden gegenereerd door een initialisatiefunctie die eenmalig wordt uitgevoerd na een reset van de microcontroller.

Bij de biphasemark code moet er een overgang (flank) zijn tussen elk paar codebits die een databit representeren. Om dit te garanderen worden de 16-bit codewoorden uit de opzoektabels al dan niet geïnverteerd, afhankelijk van het laatste bit van het voorgaande codewoord.

De opzoektabels hebben nog een tweede functie: het omkeren van de bitvolgorde. In de S/PDIF-standaard worden de subframes verstuurd beginnend met het LSB, maar de SPI doet dat net omgekeerd (MSB eerst).

Daarom leveren de LUT's de codebits in omgekeerde volgorde zodat ze door de SPI-hardware correct worden verzonden.

De op deze manier verkregen 64-bit S/PDIF-codewoorden worden in een ringbuffer (in het SRAM-geheugen van de controller) geplaatst en per DMA via SPI naar buiten gevoerd. De SPI-uitgang is rechtstreeks op de elektrische en optische S/PDIF-uitgangen aangesloten.

De S/PDIF-bitklok

Een programmeerbare fractionele frequentiedeler die in de microcontroller is geïntegreerd, leidt de S/PDIF-bitklok uit een intern 96MHz-kloksignaal af. De vereiste S/PDIF-klokfrequentie bedraagt

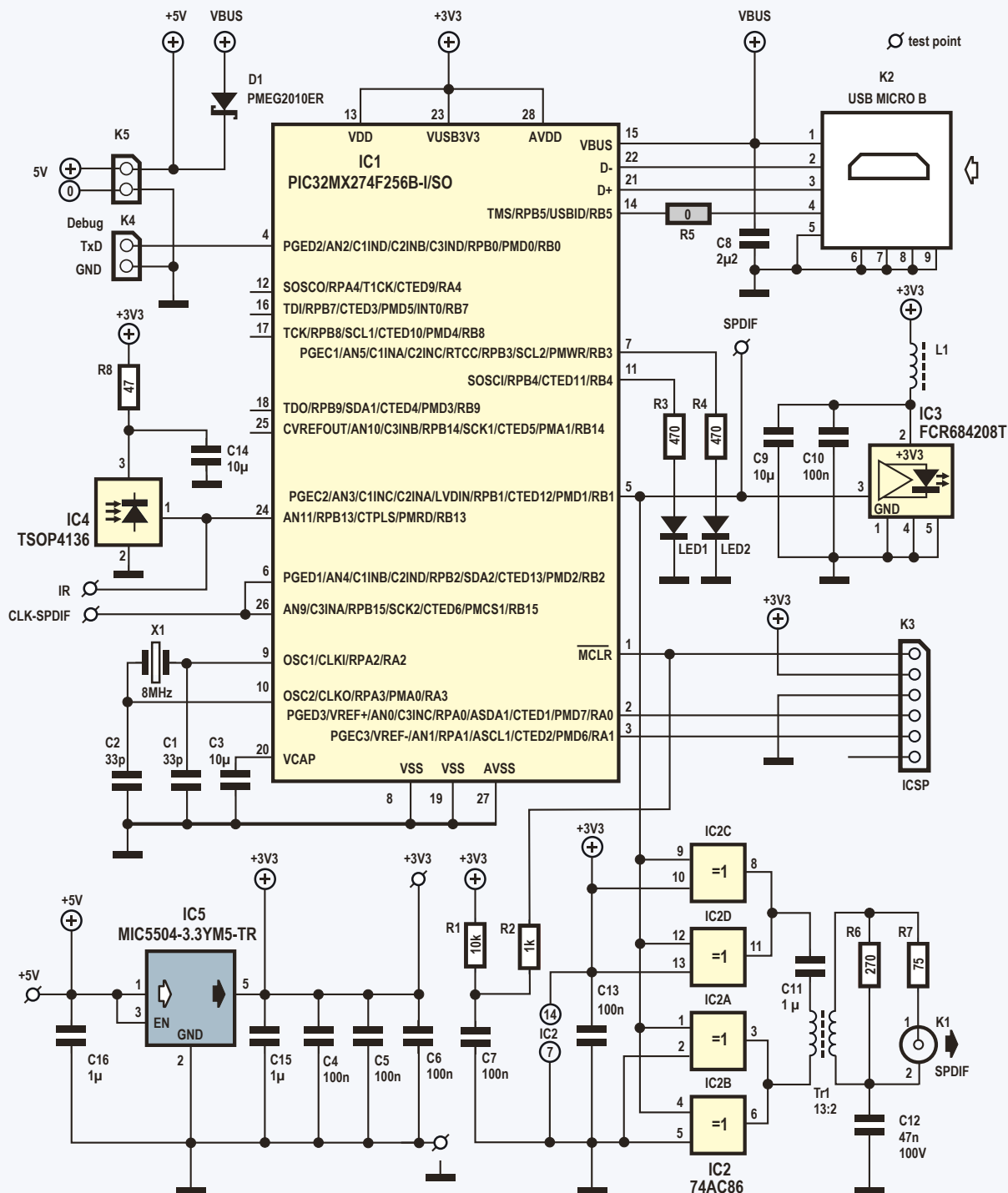
$$f_{\text{SPDIF}} = 128 * f_s$$

Hierin is f_s de audio-bemonsteringsfrequentie.

De USB-audiospecificatie geeft niet uitdrukkelijk de bemonsteringsfrequentie f_s die door de host wordt gebruikt, aan het USB-apparaat door. Dat betekent dat de μC -software deze frequentie detecteert aan de hand van het aantal PCM-frames in de isosynchrone USB-pakketten. Merk op dat de USB-configuratiedescriptor drie toegestane bemonsteringsfrequenties bevat. De host mag dus geen andere bemonsteringsfrequenties gebruiken dan deze 44,1 kHz, 48 kHz of 96 kHz. Een voordeel is dat de microcontroller eenvoudig de door de host gekozen bemonsteringsfrequentie kan 'raden' aan de hand van het aantal PCM-frames in een USB-pakket.

Klokperikelen

In overeenstemming met de USB Basic Audio Devices-specificatie [4] is voor het



180027-003-94

Figuur 5. Het uitgewerkte schema van de USB-S/PDIF-interface.

isochrone audio-eindpunt het 'synchrone' synchronisatietype gekozen. Of anders uitgedrukt: de bemonsteringssnelheid van de audiostream hangt af van het USB-klok-domein, dus van de frequentie van de SOF-tokens die door de host worden uitgevoerd. De S/PDIF-bitklok wordt daarentegen afgeleid van de lokale kristaloscillator. Om over- of leeglopen van de ringbuffer te voorkomen, past de software regelmatig het

fractionele deel van de frequentie-deelfactor aan, op basis van het gemiddelde vulniveau van de ringbuffer.

Hoewel Microchip in zijn Application Note AN1422 [10] uitdrukkelijk opmerkt dat de referentieklok tijdens bedrijf kan worden bijgesteld, ondervonden we problemen bij het aanpassen van de deelfactor bij kleine waarden daarvan (dus wanneer een bemonsteringsfrequentie van 96 kHz is geselecteerd).

Telkens wanneer de deelfactor werd aangepast, trad er een korte onderbreking in het kloksignaal op en dat leidde tot hoorbare artefacten.

Om dit probleem op te lossen hebben we een externe verbinding aangebracht tussen de uitgang van de referentie-klokoscillator REFCLKO (pen 6) en de SPI-klokingang SCK (pen 26). Op die manier omzeilen we de baudrate-generator van de SPI.



ONDERDELENLIJST

Weerstanden (alle SMD 0603, 1%, 0,1 W)

R1 = 10 k Ω
 R2 = 1 k Ω
 R3,R4 = 470 Ω
 R5 = 0 Ω niet monteren
 R6 = 270 Ω
 R7 = 75 Ω
 R8 = 47 Ω

Condensatoren (alle SMD 0603)

C1,C2 = 33 pF, 50V, 5%, C0G/NP0
 C3,C9,C14 = 10 μ F, 16V, 20%, X5R
 C4...C7,C10,C13 = 100 nF, 50V, 10%, X7R
 C8 = 2,2 μ F, 10V, 10%, X7R
 C11,C15,C16 = 1 μ F, 50V, 10%, X5R
 C12 = 47 nF, 100V, 10%, X7R

Spoeien

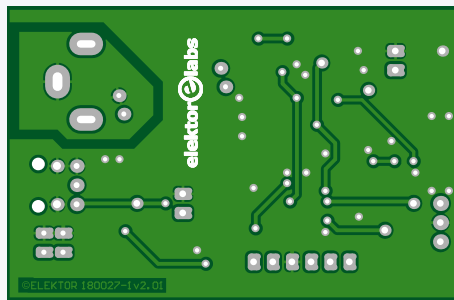
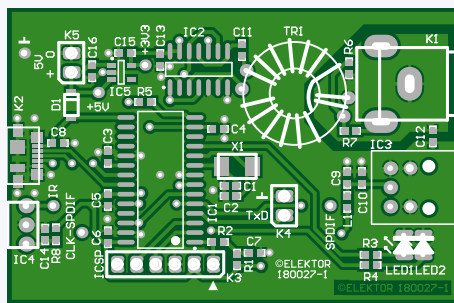
L1 = 600 Ω @ 100 MHz, 0,15 Ω , 1,3 A,
 SMD 0603 (Murata, BLM18KG601SN1D)
 TR1 = ringkern, 12,5x5mm, materiaal T38,
 Epcos B64290L0044X038

Halfgeleiders

LED1,LED2 = LED, groen, 3 mm, through-hole
 D1 = PMEG2010ER, SMD SOD-123
 IC1 = PIC32MX274F256B-I/SO, SMD SO-28
 IC2 = 74AC86, SMD SO-14
 IC3 = FCR684208T, Toslink
 (Cliff Electronic Components)
 IC4 = TSOP4136
 IC5 = MIC5504-3.3YM5-TR, SMD SOT-23-5

Overig

K1 = cinch-bus voor printmontage,
 through-hole (Pro Signal, PSG01545)
 K2 = micro-USB type B, female, SMD
 (Molex, 47346-0001)



Figuur 6. Voor de schakeling is een compact printje ontworpen.

K3 = 1x6 pinheader, verticaal, raster 2,54 mm,
 through-hole
 K4,K5 = 1x2 pinheader, verticaal, raster 2,54
 mm, through-hole
 X1 = kristal 8 MHz, 5x3,2mm, SMD
 (Abracon, ABM3-8.000MHZ-D2Y-T)
 TR1 = 0,3 m koperlakdraad (CuL), $\varnothing$ 0,5 mm
 Behuizing Hammond 1455D601,
 60 x 42,5 x 23 mm
 print 180027-1 v2.01 (Elektor-store)

IR-afstandsbediening

De IR-afstandsbedieningontvanger is geheel onafhankelijk van het audiodeel van de schakeling. Met andere woorden: alles wat door de receiver wordt ontvangen wordt rechtstreeks naar de host doorgesluisd zonder dat dit het audiodeel rechtstreeks beïnvloedt. Voor het decoderen van het IR-sigitaal van de afstandsbediening is gebruik gemaakt van de open source IRMP-bibliotheek [11]. Hiermee kunnen veel verschillende IR-signalen worden gedecodeerd. De μ C-software omvat een USB HID-implementatie [12] die het mogelijk maakt codes te versturen zoals gespecificeerd in het HID Usage Tables-document [13]. De verbinding tussen IRMP en de HID-implementatie wordt gevormd door een keymap-tabel, die de door de IRMP gedetecteerde codes koppelt aan de usage-ID's zoals gespecificeerd in [13]. We hebben besloten codes

conform de RC5-standaard toe te voegen. Omdat die steeds minder gebruikt wordt maar nog steeds voorhanden is op veel universele afstandsbedieningen, denken we dat dit in veel gevallen zal werken zonder andere apparaten zoals TV's en dergelijke te storen. In [14] zijn enkele van de RC5-codes opgesomd. Daarnaast zijn in de keymap-tabel codes voor een Denon-afstandsbediening opgenomen. De keymap staat in het bronbestand `irhid.c` onder de naam `irhid_keymap`, en kan eenvoudig aan uw wensen worden aangepast door regels toe te voegen of te verwijderen, en vervolgens de software opnieuw te compileren. Om een toets toe te voegen moet u de IRMP-codes voor het betreffende protocol, het adres en het commando zien te vinden. Dat kan door een terminal-emulator (115200 bit/s) op de S/PDIF-uitgang aan te sluiten en de betreffende toets in te

drukken. Op de terminal-emulator zou u dan de codes moeten zien. Als dat niet lukt, moet u wellicht de ondersteuning voor de betreffende afstandsbedienings-code activeren door het bestand `irmpconfig.h` te wijzigen. Daarnaast hebt u het betreffende USB HID usage-ID nodig waaraan de toets moet worden gekoppeld. Dit ID kunt u in [13] opzoeken.

Bouw

Na alle bovenstaande uitleg kunnen we over het uitgewerkte schema van **figuur 5** kort zijn. IC1 is het hart van de schakeling: de micro-controller. Deze kan in-circuit worden geprogrammeerd via ICSP-connector K3; als u daar geen zin in hebt kunt u in de Elektor-shop een geprogrammeerde controller bestellen. Voor de voeding van de schakeling is een externe +5 V-spanning nodig die op K5 wordt aangesloten.

Het USB-sigitaal komt binnen op K2 en gaat direct naar de μ C. 0-ohm-weerstand R5 moet niet worden gemonteerd omdat in de huidige software-versie USBID niet wordt gebruikt. IC4 is een standaard IR-ontvanger; IC3 verzorgt de optische (Toslink-)uitgang; via de snelle viervoudige buffer IC2 komt het elektrische uitgangssigitaal op K1 terecht. En zo komen we bij het 'merkwaardigste' onderdeel van deze schakeling: transformator TR1. Deze moet niet zozeer voor galvanische scheiding zorgen, maar moet aardlussen voorkomen. Om HF-ruis te minimaliseren moet de uitgang ontkoppeld worden – vandaar de aanwezigheid van C12. De EXOR-poortjes van IC2 fungeren als een hele brug, met als voordeel een hogere primaire spanning en een betere koppeling naar de secundaire wikkeling. De wikkerverhouding van de trafo bedraagt 13:2 en dat resulteert in een spanning van vrijwel exact 0,5 V_{tt} over een belasting van 75 Ω . Voor de schakeling is het (dubbelzijdige) printje van **figuur 6** ontworpen. Hoewel de meeste onderdelen van het SMD-type zijn, zal het volbouwen geen al te groot probleem zijn voor een redelijk geoefend soldeerartiest. Het printje past in een aluminium Hammond-kastje type 1455D601 (60 x 42,5 x 23 mm) – het kan gewoon op zijn plaats worden geschoven, zoals te zien in **figuur 7**. Afhankelijk van eventuele fabricagetoleranties kan het nodig zijn de randen van het printje een beetje bij te vijlen. Wanneer u de Hammond-behuizing uit de onderdelenlijst gebruikt, kunt u het beste de beide zwart plastic sierringen niet monteren: vooral de micro-USB connector komt dan zo diep in de behuizing dat de corresponderende stekker geen goed contact meer maakt. In de zijpaneeltjes moeten natuurlijk

passende openingen voor de connectoren en de LED's worden geboord en gezaagd. Over die LED's gesproken: de aansluitdraden daarvan worden zo gebogen dat de boven elkaar naar buiten 'kijken' – in figuur 7 is duidelijk te zien wat de bedoeling is.

Nog een opmerking over de trafo: er valt niet aan te ontkomen, die zult u zelf moeten wikkelen op de ringkern die in de onderdelenlijst is vermeld. Zoals te zien in **figuur 8** wordt de primaire wikkeling (13 windingen) als het ware in twee helften op de ringkern gewikkeld. Dat heeft het voordeel dat de aansluitingen van de primaire en de secundaire wikkelingen tegenover elkaar liggen. De secundaire bestaat uit slechts 2 windingen, en daar zal zelfs de grootste spoelhater geen moeite mee hebben. Als wikkeldraad gebruikt u 0,5 mm koperlakdraad; waarschijnlijk hebt u aan een centimeter of dertig wel genoeg.

Enkele praktische opmerkingen

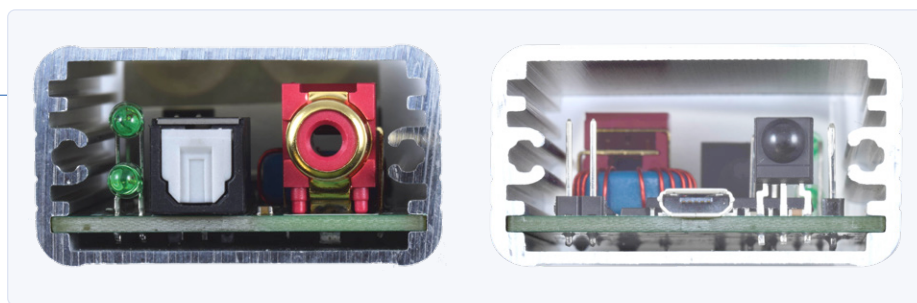
We hebben de USB-S/PDIF-interface in ons lab natuurlijk aan de tand gevoeld en met verschillende besturingssystemen uitgetoetst; dit ging met goed gevolg bij Windows 7 en Windows 10, Linux (Lubuntu en Kubuntu), Android en zelfs Raspbian op een Raspberry Pi 3 model B+ onder gebruikmaking van *2019-09-26-raspbian-buster-full.img*.

Windows

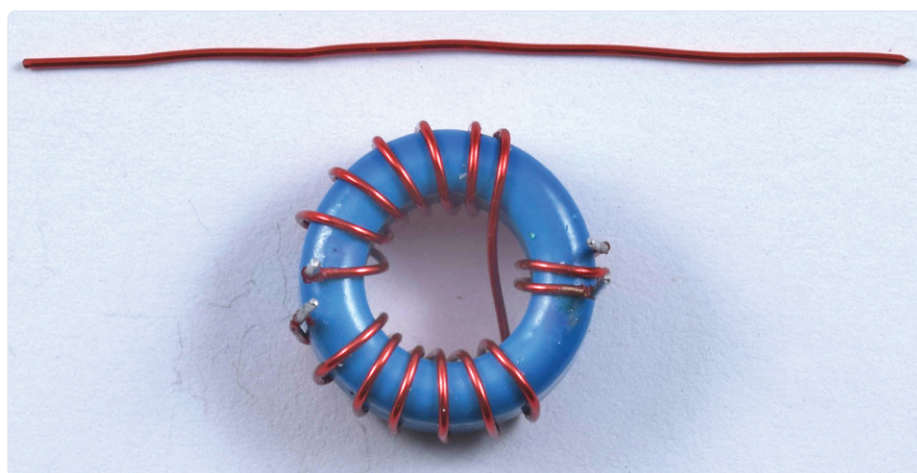
Windows herkent de interface automatisch; de bemonsteringssnelheid kan worden ingesteld in het Control Panel onder de optie Sound. Selecteer 'SPDIF Interface Properties'; op het tabblad Advanced kunt u het formaat kiezen, zoals in **figuur 9**.

Lubuntu

Bij Linux (wij gebruikten Lubuntu) hangen de bemonsteringsfrequenties van het bestand af. Onder gebruikmaking van de eenvoudige GNOME MPlayer verschijnt een 44,1kHz-bestand met 44,1 kHz op de S/PDIF-uitgang. Maar we moeten telkens 'USB_SPDIF Stereo (IEC958) (PulseAudio)' als Audio Output selecteren – ook als we dat eerder ook al hadden gedaan. Dat is bijzonder storend. Om dat te omzeilen kunt u alle andere audio-apparaten uitschakelen (hier: Sound & Video – PulseAudio Volume Control – Configuration). Een 48kHz-bestand en een 96kHz-bestand worden beide met 48 kHz afgespeeld. Een andere mogelijkheid is om ALSA rechtstreeks te gebruiken zonder PulseAudio in een terminal. Wanneer u niets hoort, start u eerst als mixer in een terminal (Ctrl+Alt+T en voer `alsamixer` in); druk dan op F6 om de



Figuur 7. De opgebouwde print kan in de aluminium behuizing worden geschoven.



Figuur 8. Zo wordt de ringkerntrafo gewikkeld. Merk op hoe de primaire wikkeling als het ware in twee delen is gesplitst.

geluidskaart 'USB_SPDIF' te selecteren (er verschijnt slechts één klein besturings-element dat 00 en PCM toont, omdat de interface geen volumeregeling heeft). Sluit de mixer dan af en voer in (//.../ staat voor het pad naar het audiobestand:

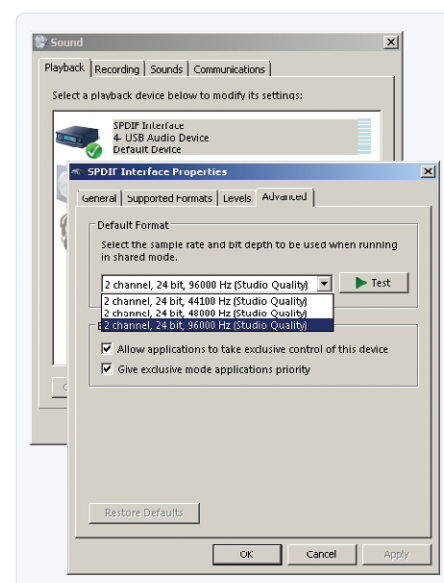
```
aplay -D plughw:CARD=USBSPDIF //.../
```

Zo kunt u alleen niet-gecomprimeerde wav-bestanden afspelen. Om mp3-bestanden vanaf de commandoregel af te spelen, kunt u bijvoorbeeld mpg123 gebruiken, maar er zijn nog talloze andere mogelijkheden. Hoewel er al vele jaren USB Audio-DAC's bestaan, is de standaardmanier waarop Linux audiobestanden behandelt nog altijd voor verbetering vatbaar. Het instellen van de uitgangsbemonsteringsfrequentie is ook mogelijk door het configuratiebestand van PulseAudio aan te passen (`/etc/pulse/daemon.conf`). Informatie hierover kan online worden gevonden. Maak echter van tevoren wel een backup van het originele bestand – een ongeluk zit in een klein hoekje...

Android

Sluit de interface aan met een OTG-kabeltje. Installatie van een app met de naam

'USB Audio Player Pro' is waarschijnlijk de beste manier om onze interface te gebruiken. Daarmee worden de audio-beperkingen van Android omzeild. Alle drie bemonsteringsfrequenties van de interface worden ondersteund. Nadat de app gestart is, wordt de interface meteen herkend.



Figuur 9. Instellingen onder Windows 7.



SHOPPING LIST

- > **USB-S/PDIF-interface, kale print:** www.elektor.nl/180027-1
- > **USB-S/PDIF-interface, geprogrammeerde controller:** www.elektor.nl/180027-41

Raspbian

Als laatste hebben we de interface in combinatie met Raspbian uitgetest. Net als Ubuntu is Raspbian een gratis besturings-systeem dat ook op Debian is gebaseerd; het hoeft dus geen verbazing te wekken dat de test hetzelfde patroon volgde. Het afspelen van niet-gecomprimeerde wav-bestanden gaat dus zo:

```
aplay -D plughw:CARD=USBSPDIF //.../
```

Om het afspelen te stoppen voert u Ctrl+c in.

Bij het afspelen van mp3-bestanden vanaf de commandoregel, werkte mpg123 (als dat niet geïnstalleerd is: `sudo apt-get install`

mpg123) bij onze verse Raspbian-installatie pas nadat PulseAudio en diens volumeregeling waren geïnstalleerd. Zo installeert u PulseAudio:

```
sudo apt-get install pulseaudio
```

Installatie van de bijbehorende volumeregeling:

```
sudo apt-get install pavucontrol  
paprefs
```

Start het systeem na installatie opnieuw op. Net als bij Ubuntu moet u de onboard-audio uitschakelen in Sound & Video – PulseAudio Volume Control – Configuration.

De voorgeïnstalleerde VLC Media Player produceerde pas geluid nadat ook PulseAudio was geïnstalleerd. Een met 32 kHz bemonsterd bestand wordt met 96 kHz afgespeeld. Natuurlijk hebben we ook de audioprestaties gemeten. Voor een beschrijving van deze metingen en de uitkomsten verwijzen we naar de Elektor Labs-pagina van dit project [15]. De op het moment van schrijven actuele software kan worden gedownload van de projectpagina bij dit artikel [16] of van GitHub [17].

Rest ons slechts u veel bouw- en luisterplezier te wensen! Als altijd zijn uw op- en aanmerkingen welkom. 

180027-01

WEBLINKS

- [1] https://kompendium.infotip.de/spdif_toslink.html
- [2] **Universal Serial Bus Specification Revision 2.0:** www.usb.org/document-library/usb-20-specification
- [3] **Universal Serial Bus Device Class Definition for Audio Devices, Release 1.0:** www.usb.org/document-library/audio-device-document-10
- [4] **Universal Serial Bus Audio Device Class Specification for Basic Audio Devices, Release 1.0:** www.usb.org/document-library/audio-device-class-spec-basic-audio-devices-v10-and-adopters-agreement
- [5] **Universal Serial Bus Device Class Definition for Audio Data Formats, Release 1.0:** www.usb.org/document-library/audio-data-formats-10
- [6] **Universal Serial Bus Device Class Definition for Terminal Types, Release 1.0:** www.usb.org/document-library/audio-terminal-types-10
- [7] AES3-2003: AES standard for digital audio – Digital input-output interfacing – Serial transmission format for two-channel linearly represented digital audio data
- [8] **Crystal Application Note AN22: "Overview Of Digital Audio Interface Data Structures":** <https://statics.cirrus.com/pubs/appNote/an22.pdf>
- [9] **Sean Eron Anderson: "Bit Twiddling Hacks", section "Compute parity in parallel":** <http://graphics.stanford.edu/~seander/bithacks.html#ParityParallel>
- [10] **Microchip AN1422: "High-Quality Audio Applications Using the PIC32":** <http://ww1.microchip.com/downloads/en/AppNotes/01422A.pdf>
- [11] **Infrared Multi-Protocol decoder (IRMP):** www.mikrocontroller.net/articles/IRMP
- [12] **USB Device Class Definition for Human Interface Devices (HID), Version 1.11:** www.usb.org/document-library/device-class-definition-hid-111
- [13] **USB HID Usage Tables, Version 1.12:** www.usb.org/document-library/hid-usage-tables-112
- [14] **"IR-afstandsbedieningscodes, deel 1", in Elektor maart 2001:** www.elektormagazine.nl/magazine/elektor-200103/13670
- [15] **Projectpagina op Elektor Labs:** www.elektormagazine.com/labs/usb-spdif-interface-180027
- [16] **Projectpagina bij dit artikel:** www.elektormagazine.nl/180027-01
- [17] **Software-download op GitHub:** <https://github.com/kiffie/usb-spdif>