



# **BAUANLEITUNG**

## **FREEDSP-AURORA**

**DIY**

A large, red, rectangular stamp with the word "DIY" written in a bold, distressed, sans-serif font. The stamp is tilted slightly to the right.



## REVISIONSPROTOKOLL

Revision	Beschreibung	Datum
v1.0	Erste Version	28 Nov 2018
v1.1	Hardware release	04 Mai 2019



Revisionsprotokoll	2
About freeDSP-aurora	4
Wichtige Information	6
Features	7
Übersicht	8
Blockdiagramm	8
Anschlüsse	9
Optionale Eingangs- und Ausgangs-AddOns	12
freeDSP-aurora aufbauen	13
Bauteile und WerkzeugE	13
Löten der Platine	14
ESP32 ProgrammiEREN	15
XMOS XE216-512-TQ128 programmieren	18
Hochladen eines DSP-Plugins	25
Anhang	26
Stückliste	26
BestückungsplaN	33
Schaltpläne	34



## ABOUT FREEDSP-AURORA

FreeDSP-aurora ist eine kostengünstige Lösung für Audiosignalverarbeitung in Echtzeit für Forscher und für die DIY-Community. Du kannst die Platine in Deine eigenen Projekte integrieren. Die Platine wird ohne Gehäuse geliefert. Einfacher Zusammenbau und einfache Programmierung stehen im Mittelpunkt. FreeDSP-aurora basiert auf dem ADAU1452 DSP von Analog Devices zusammen mit der kostenlosen Programmierumgebung SigmaStudio. Die Programmierung erfolgt mit Funktionsblöcken ähnlich wie in anderen grafischen Programmierumgebungen wie PureData oder Max/MSP. Vorgefertigte Funktionsblöcke (z.B. Filter, Kompressoren, Effekte oder Logik) können per Drag&Drop in den Signalweg eingefügt werden. Falls eine Funktion in der Bibliothek nicht vorhanden ist, kann diese aus Low-Level-Blöcken (Multiplizierer, Delays) aufgebaut werden. Auf der Analog Devices Website findest Du weitere Informationen hierzu.

FreeDSP-aurora bietet viele Möglichkeiten an digitaler Signalverarbeitung und Einstellungen. FreeDSP-aurora kann für verschiedene Audioanwendungen eingesetzt werden, z.B.:

Raum-/Systementzerrung

Digitale Frequenzweiche in aktiven Lautsprecherkonzepten

Multiband Dynamics Processing

Delay Kompensation / Phasendrehung

Bassenhancement

Subwooferintegration

Audioeffekte für Musikinstrumente

Verbesserung der Stereoabbildung

...

Ein XMOS XE216-512-TQ128 MCU stellt eine USB Audio Class 2 kompatible Schnittstelle für einen Computer mit macOS, Linux oder Windows 10 bereit. Es stehen 8 Audioein- und 8 Audioausgänge zur Verfügung. Zusätzlich stellt die XMOS MCU einen ADAT Ein-/Ausgang und einen Wordclock Ein-/Ausgang bereit.



Die ESP32 MCU steuert den DSP, stellt WiFi oder Bluetooth her und verarbeitet Ein- und Ausgaben der Peripherie wie Drehenkoder, Display, Temperatursensor, PWM-Lüfter und Infrarot-Empfänger.

Als Teil des Projektes wurde eine OpenSource-Software veröffentlicht, die freeDSP-aurora von macOS, Windows oder iOS aus steuert. Mit dieser Software erhältst Du Zugriff auf die Parameter des installierten DSP-Plugins. Dank der OpenSource-Lizenz kannst Du die Steuersoftware jederzeit modifizieren, um Deine selbsterstellten DSP-Plugins zu steuern.

Die Pläne und die Software für freeDSP-aurora werden unter einer Creative Commons Attribution ShareAlike 4.0 International (CC BY-SA 4.0) Lizenz veröffentlicht. Diese erlaubt die uneingeschränkte Nutzung und Modifikation der Hard- und Software. Erfahrene Benutzer können ihre eigenen Versionen der Platine erstellen, diese erweitern oder verbessern, solange freeDSP und auverdion als ursprüngliche Entwickler benannt und ihr Design unter der gleichen Lizenz veröffentlichten.

Der freeDSP-Name und das freeDSP-Logo sind Copyright von Sebastian Merchel und Ludwig Kormann und dürfen nicht ohne schriftliche Genehmigung benutzt werden.

Der Name auverdion ist Copyright von Raphael Knoop und darf nicht ohne schriftliche Genehmigung benutzt werden.

Diese Anleitung wird unter der gleichen CC-Lizenz veröffentlicht.



## WICHTIGE INFORMATION

Das freeDSP-aurora.Modul kann Signale generieren, die Dein Audioequipment beschädigen können. Bitte lies die vorliegende Anleitung sorgfältig bevor Du mit freeDSP-aurora arbeitest. Nimm alle Hardware- und Softwarekonfigurationen vor, bevor Du weiteres Audioequipment anschließt. Beginne immer mit niedriger Lautstärke und erhöhe sie nur langsam, um das Risiko einer Beschädigung Deines Audiosystems gering zu halten.

FreeDSP-aurora wird Dir so, wie beschrieben, zur Verfügung gestellt. Wir geben keinerlei ausdrückliche oder stillschweigende Garantien in Bezug auf Funktionalität, Funktionsfähigkeit oder Verwendung ab, einschließlich, ohne darauf beschränkt zu sein, implizite Garantien der Marktgängigkeit, der Eignung für einen bestimmten Zweck oder einer Verletzung. Wir lehnen ausdrücklich jegliche Haftung für direkte oder indirekte Folgeschäden, zufällige oder besondere Schäden ab, einschließlich - ohne Einschränkung - entgangener Einnahmen, entgangener Gewinne, Verluste durch Betriebsunterbrechung oder Datenverlust, unabhängig von der Handlungsweise oder der Rechtslehre, unter der sich die Anwender befinden, auch wenn auf die Möglichkeit oder die Wahrscheinlichkeit eines solchen Schadens hingewiesen wird. Merkmale und Spezifikationen können ohne vorherige Ankündigung geändert werden.

Bitte beachten Sie, dass freeDSP-aurora ein Open-Source-Freizeitprojekt ist. Da freeDSP-aurora sehr flexibel ist, sind viele Anwendungen möglich. Fragen und neue Ideen können online mit anderen DIYlern besprochen werden. Bitte verwende dazu das Digital Line Level-Unterforum @ diyAudio.com oder das Elektronik-Unterforum @ www.diy-hifi-forum.eu, um Dich mit anderen Personen zu treffen, die mit freeDSP-aurora arbeiten. Erstelle neue Threads nur dann, wenn Du das Problem nicht in den vorhandenen Threads finden konntest. Einige Fragen können durch sorgfältiges Lesen dieser Anleitung beantwortet werden. Wir können keinen individuellen Support per E-Mail anbieten. Danke für Dein Verständnis!

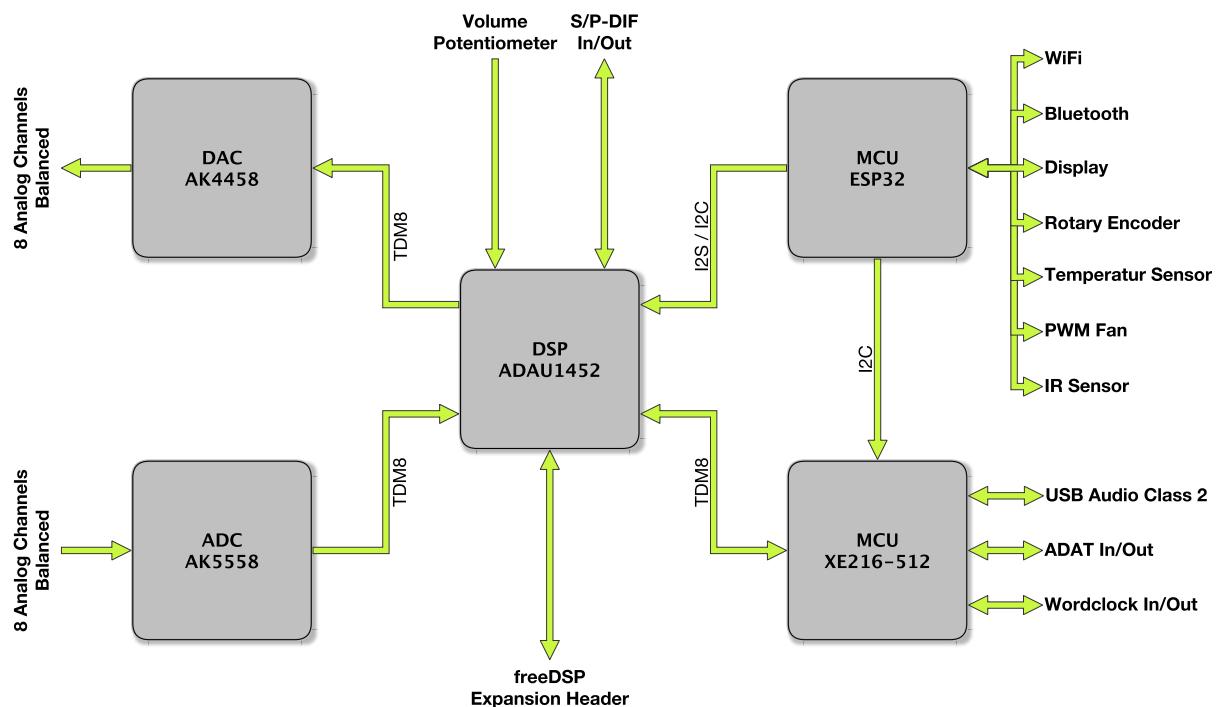


## FEATURES

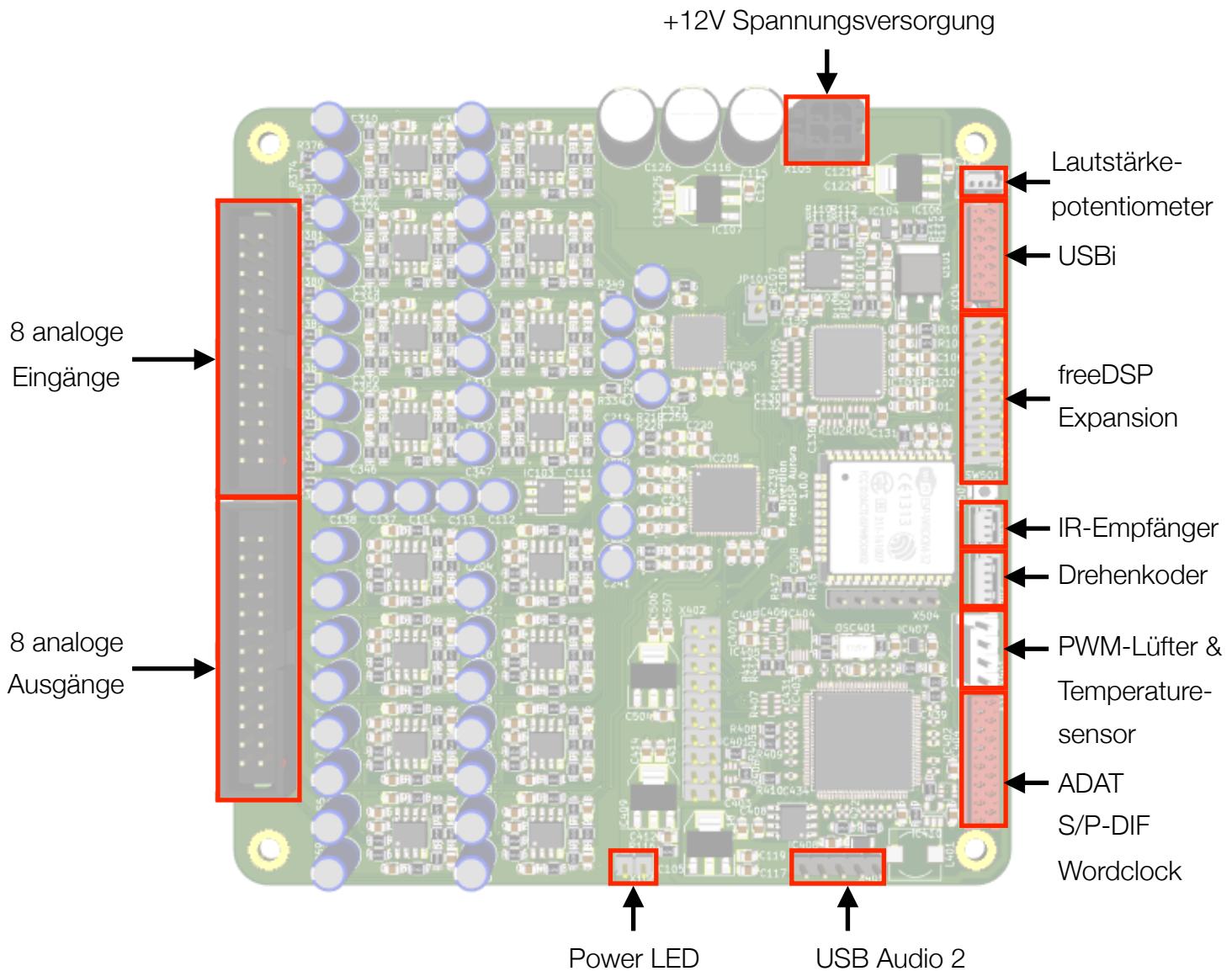
- Analog Devices ADAU1452, 294.912 MHz, 32-bit SigmaDSP
  - 6144 SIMD Instruktionen pro Abtastwert @ 48kHz fs
  - 40kWords Data RAM
  - 800ms Digital Audio Delay Pool @ 48kHz fs
  - 8 Stereo ASRCs mit 139dB DNR
- XMOS XE216-512-TQ128 für multikanal-bidirektionales Audiostreaming
- ESP32 für Steuerung über WiFi oder Bluetooth
- AKM AK4458 32bit-DAC
- AKM AK5558 32bit-ADC
- Unterstützte Abtastraten zwischen 44.1kHz und 192kHz
- 8 Analoge Eingänge, symmetrisch, +6dBu
- 8 Analoge Ausgänge, symmetrisch, +6dBu
- S/P-DIF Ein- und Ausgang
- ADAT Ein- und Ausgang
- Wordclock Ein- und Ausgang
- Unterstützung für Display, Drehenkoder, Lautstärkepotentiometer, Temperatursensor, PWM-Lüfter, IR-Sensor
- Ein freeDSP-Erweiterungssport
- USB Audio Class 2 bidirektionales Audiostreaming mit 8 Eingangs- und 8 Ausgangskanälen, full-duplex. Unterstützt von ASIO-Treibern unter Windows 10, von macOS und Linux ohne Treiberinstallation unterstützt.
- Steuersoftware für Windows, macOS, iOS über WiFi, veröffentlicht unter OpenSource Lizenz
- Abmessungen: 110mm x 110mm

# ÜBERSICHT

## BLOCKDIAGRAMM



## ANSCHLÜSSE





### Analoge Audioanschlüsse

Analoge Audioeingänge kannst Du am Stecker X201 anschliessen. Die Eingänge sind für symmetrischen Betrieb bei +6dBu maximaler Eingangsspannung ausgelegt. Unsymmetrische Quellen können angeschlossen werden, indem der negative Eingang mit Masse verbunden wird. In diesem Fall sollte überprüft werden, ob die Verstärkung der Eingangsstufe angepasst werden muss. Alternativ kann eine Schaltung zur Wandlung von unsymmetrisch auf symmetrisch oder zur Anpassung verschiedener Signalpegel verwendet werden.

Analoge Audioausgänge kannst Du am Stecker X301 anschliessen. Die Ausgänge sind für symmetrischen Betrieb bei +6dBu maximaler Ausgangsspannung ausgelegt.

Unsymmetrische Empfänger können angeschlossen werden, indem der negative Eingang mit nicht verbunden wird. In diesem Fall sollte überprüft werden, ob die Verstärkung der Ausgangsstufe angepasst werden muss. Alternativ kann eine Schaltung zur Wandlung von symmetrisch auf unsymmetrisch oder zur Anpassung verschiedener Signalpegel verwendet werden.

Wenn Du Audioverbindungen herstellst, schalte zuvor das gesamte Audioequipment aus, um Schäden zu vermeiden.

Die Verstärkung der Ein- und Ausgänge kann nur durch Ändern der Widerstände auf der Platine angepasst werden.

### Digitale Audioverbindungen

Digitale Audiosignale wie S/P-DIF Ein- und Ausgang oder ADAT Ein- und Ausgang kannst Du am Stecker X101 anschliessen. Dort können auch der Wordclock-Ein- und Ausgang angeschlossen werden.

### FreeDSP Expansion Header

An der Stiftleiste X102 können Boards angeschlossen werden, um freeDSP-aurora mit zusätzlichen Ein- und Ausgängen zu erweitern. Die Anschlußbelegung entspricht dem I2S-Expansion-Header des freeDSP-Projekts. Für ein I2C-Display ist X102 ebenfalls vorgesehen.

### Lüfter und Temperatursensor

Einen PWM-gesteuerten Lüfter kannst Du an X501 anschliessen, Zusätzliche kann am Sense-Pin ein Temperatursensor (z.B. NZC) angeschlossen werden.



### Drehenkoder

Einen Drehenkoder (mit oder ohne Taster) kannst Du an X502 anschliessen.

### IR-Empfänger

An X503 kannst Du einen Infrarot-Empfänger für eine Fernbedienung anschliessen.

### USB-Anschluss

Einen Computer kannst Du an der Stiftleiste X401 anschliessen. Dafür wird ein Adapterkabel aus dem Computerzubehör benötigt. Vergewissere Dich, daß die Pinbelegung des Kabels mit der Pinbelegung der Steckerleiste übereinstimmt, um Beschädigungen am Computer oder am freeDSP-aurora zu vermeiden. Üblicherweise sind die Massepins 4 & 5 durch eine schwarze Aderisolierung gekennzeichnet, es gibt aber auch Ausnahmen, weshalb Du dies doppelt überprüfen solltest.

FreeDSP-aurora entspricht dem USB Audio Class 2 Standard. Daher wird für macOS und Linux keine Treiberinstallation benötigt, da diese Betriebssysteme UAC2-Audiointerfaces nativ unterstützen. Windows 10 wird inzwischen ebenfalls mit einem Treiber für UAC2 ausgeliefert. Dennoch kann es notwendig sein, zusätzlich die freie Software asio4all zu installieren und Software mit ASIO-Unterstützung zu verwenden. Dies ist dann der Fall, wenn Du mit freeDSP-aurora 8 Audiokanäle am Computer aufnehmen willst. Wenn Du lediglich am Computer 8 Kanäle über freeDSP-aurora wiedergeben möchtest, ist ASIO nicht zwingend notwendig.

### Power LED

An diesem Anschluss kann eine LED angeschlossen werden, um den Einschaltzustand anzuzeigen.

### +12V-Spannungsversorgung

FreeDSP-aurora muss mit einer Spannungsversorgung von +12V DC an X105 versorgt werden. Achtung: Schalte die Spannungsversorgung erst ein, wenn alle Verbindungen hergestellt worden sind und Du alles doppelt überprüft hast.



## OPTIONALE EINGANGS- UND AUSGANGS-ADDONS

FreeDSP-aurora wurde so entworfen, daß möglichst viele Anwendungen unterstützt werden. Daher wurden alle Ein- und Ausgänge auf Stifteleisten oder Anschlüsse für Flachbandkabel gelegt. Somit können Anwender die Anschlüsse an ihre eigenen Bedürfnisse anpassen. Für dieses Projekt werden einige AddOns verfügbar sein. Diese decken die häufigsten Anwendungsfälle ab, z.B. Aktive Mehrweglautsprecher, oder ein 8-Kanal Lautsprechermanagement, sowie Display und Schalter. Bitte besuche das github-Repository oder die Website für mehr Informationen über die AddOns.



## FREEDSP-AURORA AUFBAUEN

Die in diesem Kapitel beschriebenen Schritte sind nur dann notwendig, wenn Du die freeDSP-aurora-Hardware selbst zusammenbaust und die Firmwares programmieren willst. Wenn Du lediglich ein neues DSP-Plugin hochladen willst, kannst Du direkt zum Kapitel **Hochladen eines DSP-Plugins** weitergehen.

### BAUTEILE UND WERKZEUGE

Du benötigst einen Lötkolben mit einer feinen Bleistiftspitze und ein wenig Löterfahrung, um die SMD- und THT-Bauteile zu montieren.

Zusätzlich benötigst Du:

- einen USB2Serial-Konverter wie ein FTx232-Modul, z.B. [https://www.ftdichip.com/Products/Modules/DevelopmentModules.htm#FT2232H\\_Mini](https://www.ftdichip.com/Products/Modules/DevelopmentModules.htm#FT2232H_Mini) oder ähnlich
- optional einen XTAG-Debugadapter <https://www.xmos.com/support/boards?product=19480>, wenn Du die Features nutzen willst, die durch die XMOS-MCU bereitgestellt werden
- optional einen USBi-Adapter von Analog Devices (EVAL-ADUSB2EBZ) oder einen freeUSBi-Adapter (s. freeDSP-Website), wenn Du den DSP aus SigmaStudio heraus programmieren willst.

Bestelle einen freeDSP-aurora-Bausatz. Manchmal werden Sammelbestellungen für die Platine oder alle Bauteile in den DIY-Foren angeboten. Da Teilnehmer dies in ihrer Freizeit organisieren, kann es eine Weile dauern, bis eine neue Sammelbestellung angeboten wird.

- (a) Alternativ kannst Du die Platine über [www.tindie.com](http://www.tindie.com) bestellen. Zusätzlich musst Du die Bauteile bestellen bei Digikey (oder einem anderen Distributor) bestellen. Dies ist vermutlich die schnellste Option.
- (b) Alternativ kannst Du die Platine selbst herstellen oder herstellen lassen, wenn Du Änderungen am originalen Entwurf vorgenommen hast. Diese Option ist am flexibelsten, erfordert aber auch mehr Erfahrung (und Geld), um alles funktionsfähig aufzubauen.



- (c) Alternative kannst Du die original Platine mit den Daten auf github selber herstellen oder herstellen lassen. Du kannst Dich mit Deinen Freunden zusammentun und eine eigene Sammelbestellung organisieren. Alle notwendigen KiCAD-Dateien findest Du über den Link auf der freeDSP-Website [www.freeDSP.cc](http://www.freeDSP.cc). Die Bauteile kannst Du über DigiKey (oder einen anderen Distributor) bestellen.

## LÖTEN DER PLATINE

Beginne mit dem Löten der schwierigsten Bauteile: Die ICs. Auf youtube findest Du Tutorials, wie QFP-Gehäuse von Hand gelötet werden können, z.B.

<https://www.youtube.com/watch?v=YUryJOAiPa4>

und wie QFN-Gehäuse von Hand gelötet werden können, z.B.

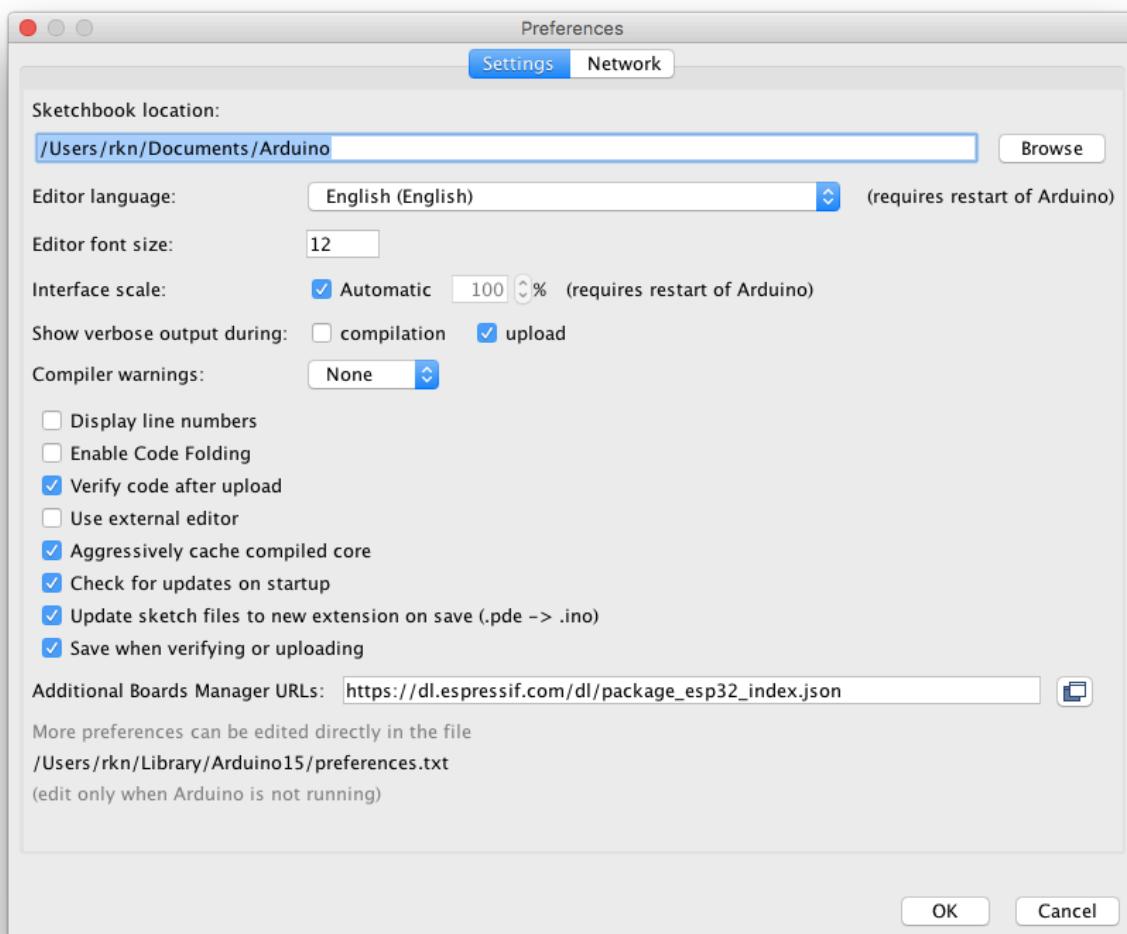
<https://www.youtube.com/watch?v=BvhE16vBfX4>

Danach lötest Du alle anderen SMD-Bauteile und zum Schluss die THT-Bauteile. Beginne immer zuerst mit den Bauteilen mit geringster Höhe.

## ESP32 PROGRAMMIEREN

Die ESP32-MCU auf dem Modul wird benötigt, um Parameteränderungen zur Laufzeit vorzunehmen, diese auf dem Modul zu speichern oder um neue DSP-Plugins zu installieren. Um die ESP32-MCU mit der aktuellsten Firmware zu programmieren, musst Du folgende Schritte ausführen:

1. Lade und installiere die Arduino IDE kostenlos von <https://www.arduino.cc>.
2. Starte die Arduino IDE, öffne den Preferences-Dialog und füge [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) für *Additional Boards Manager URLs* ein und klicke *OK*. Dann schliesse die Arduino IDE.



3. Installiere die ESP32-Werkzeuge, wie in diesem Link für Dein jeweiliges Betriebssystem beschrieben:  
<https://github.com/espressif/arduino-esp32#installation-instructions>
4. Nach erfolgreicher Installation und Neustart der Arduino IDE solltest Du einige ESP-Boards unter *Tools->Board* sehen. Bitte wähle *ESP Dev Module* aus.
5. Nun verbinde Deinen USB2Serial-Adapter mit X504: TXD (pin 3), RXD (pin 4) und GND (pin 6) werden mit den entsprechenden Pins des USB2Serial-Adapters verbunden.
6. Bitte überprüfe nochmals Pinbelegung und die Verbindungen. Dann verbinde den USB2Serial-Adapter mit einem USB-Port an Deinem Computer.
7. Wähle Deinen USB2Serial-Adapter unter *Tools->Port* und öffne den Serial Monitor der Arduino IDE.
8. Bei Einschalten der Spannungsversorgung des freeDSP-aurora-Moduls solltest Du jetzt die Bootlog-Nachrichten des ESP32 im Serial Monitor sehen. Falls nicht, überprüfe der Verkabelung, vielleicht sind RXD und TXD vertauscht.
9. Der ESP32 kann nun mit der aktuellsten Firmware programmiert werden.
10. Verbinde zunächst pin 1 und pin 2 auf der Stiftleiste X504. Wenn Du jetzt den Taster SW501 drückst, erscheint im Serial Monitor die Nachricht, daß der ESP32 auf den Download eines neuen Programms wartet.
11. Öffne die Datei aurora.ino aus dem freeDSP-aurora-Repository in der Arduino IDE.
12. Klicke *Sketch->Click/Verify*. Die Arduino IDE kompiliert nun die Firmware.  
Anschließend klicke *Sketch->Upload*, um den Upload der Firmware auf den ESP32 Deines freeDSP-aurora-Moduls zu starten.  
Achtung: Falls der Upload fehlschlägt hast Du vermutlich vergessen, den ESP32 in den Uploadmodus zu versetzen, wie es in Schritt 10 beschrieben ist.
13. Nach erfolgreichem Upload entferne die Verbindung aus Schritt 10 und starte freeDSP-aurora neu, indem Du die Spannungsversorgung aus- und wieder einschaltest. Im Wifi-Controlpanel Deines Computers erscheint freeDSP-aurora in der Liste möglicher Access-Points für Adhoc-Verbindungen.
14. Jederzeit kannst Du Änderungen an der ESP32-Firmware vornehmen und diese auf Dein freeDSP-aurora hochladen, indie Du die Schritte 10-13 wiederholst.



Nach dem ersten Hochladen einer Firmware kannst Du auch den Over-The-Air-Upload (OTA) benutzt, um eine neue Firmware über Wifi auf den ESP32 hochzuladen ohne einen USB2Serial-Adapter anschliessen zu müssen. Folge dazu den Anweisungen in diesem Link:  
[https://www.az-delivery.de/blogs/azdelivery-blog-für-arduino-und-raspberry-pi/ota-over-the-air-esp-programmieren-über-wlan-entwurf?ls=en](https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/ota-over-the-air-esp-programmieren-uber-wlan-entwurf?ls=en)

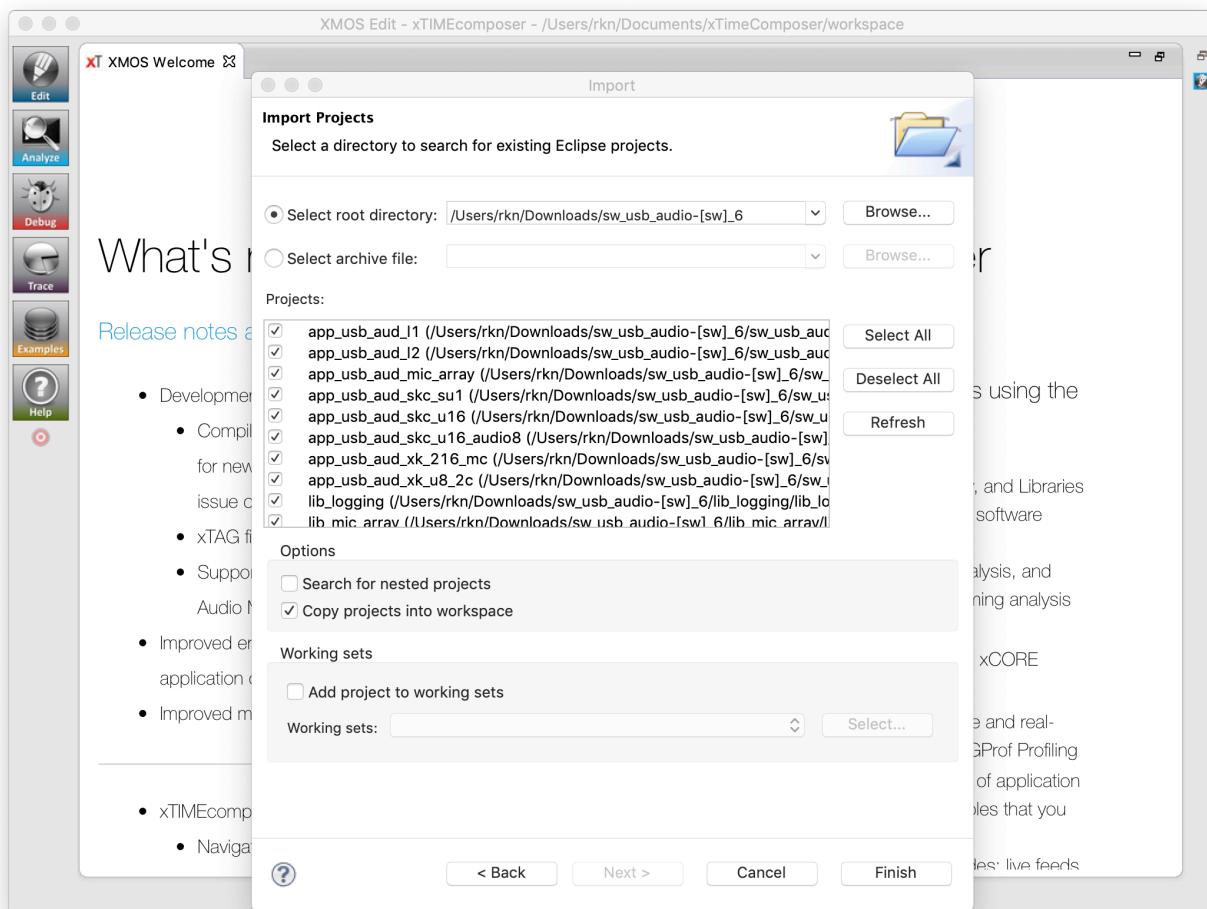
OTA kann derzeit nur verwendet werden, wenn freeDSP-aurora erfolgreich mit einem lokalen Wifi-Netzwerk verbunden wurde und Dein Computer sich im selben Netzwerk befindet. Mit einer Adhoc-Verbindung funktioniert OTA nicht.

Wenn Du OTA mit einer modifizierten Firmware verwenden willst, vergewissere Dich, daß diese OTA unterstützt.

## XMOS XE216-512-TQ128 PROGRAMMIEREN

Wenn Du die Features ADAT In/Out, Wordclock In/Out oder USB Audio Class 2 nutzen möchtest, musst Du den XE216-512-TQ128 programmieren:

1. Lade das xTimeComposer Studio kostenlos hier herunter: <https://www.xmos.com/support/tools>.
2. Lade die XMOS USB Audio 2.0 Device Software Version: 6.15.2rc1 Source Code Framework kostenlos hier herunter: <http://www.xmos.com/published/usbaudiodevice-software?version=all> und entpacke das Archiv (falls nicht automatisch geschehen).
3. Importiere das Framework in Deinen Workspace:  
Klicke *File->Import General->Existing Project into workspace* und dann *Next*.
4. Wähle das Verzeichnis aus, in das Du das Framework entpackt hast und aktiviere *Copy projects into workspace*.



5. Klicke *Finish*.
6. Im xTIMEcomposer Studio kopiere das Projekt app\_usb\_aud\_xk\_216\_mc und benenne die Kopie in app\_freeDSP-aurora um.
7. Kopiere die Dateien von <git repository>/SOURCES/XMOS/src/core nach <workspace>/app\_freeDSP-aurora/src/core. Bereits vorhandene Dateien müssen mit den Dateien aus dem Repository ersetzt werden.
8. Öffne die Datei audiohw.xc im xTIMEcomposer Studio und ändere die Funktion AudioHwConfig wie folgt:

```

void AudioHwConfig(unsigned samFreq, unsigned mClk, chanend ?
c_codec, unsigned dsdMode,
                    unsigned sampRes_DAC, unsigned sampRes_ADC)
{
    unsigned char data[1] = {0};

    /* Set master clock select appropriately */
#if defined(USE_FRACTIONAL_N)
    /* Configure external fractional-n clock multiplier for
300Hz -> mClkFreq */
    PllMult(mClk, PLL_SYNC_FREQ);
#endif
    /* Allow some time for mclk to lock and MCLK to stabilise -
this is important to avoid glitches at start of stream */
{
    timer t;
    unsigned time;
    t :> time;
    t when timerafter(time+AUDIO_PLL_LOCK_DELAY) :> void;
}

#if defined(USE_FRACTIONAL_N)
while(1)
{

```

```
/* Read Unlock Indicator in PLL as sanity check... */
CS2100_REGREAD(CS2100_DEVICE_CONTROL, data);
if (!(data[0] & 0x80))
{
    break;
}
#else
if (mClk == MCLK_441)
{
    set_gpio(P_GPIO_MCLK_FSEL, 0);
}
else
{
    set_gpio(P_GPIO_MCLK_FSEL, 1); //mClk = MCLK_48
}

/* Allow MCLK to settle */
wait_us(20000);

#endif

return;
}
```

9. Öffne die Datei customdefines.h im xTIMEcomposer Studio, gehe zu den USB-Descriptordefinitionen und ändere die Product- und Vendorstrings, z.B.

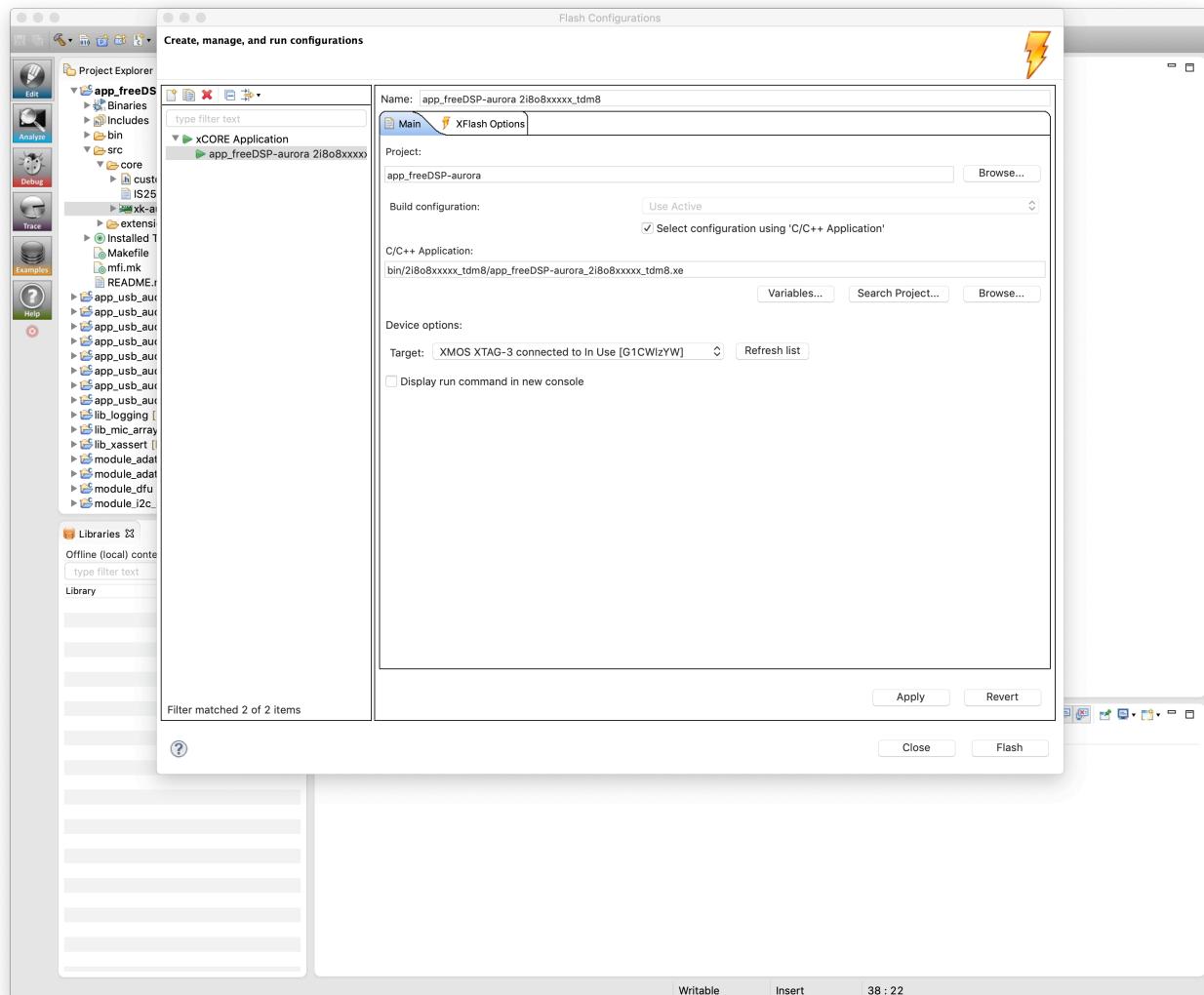
```
#define PRODUCT_STR          "freeDSP-aurora"
#define VENDOR_STR           "auverdion"
```

10. Ändere VENDOR\_ID, PID\_AUDIO\_1 und PID\_AUDIO\_2 in Deine eignene VID und PID.
11. Öffne die Datei Makefile im xTIMEcomposer Studio, suche die Zeile, die mit XCC\_FLAGS\_2i808xxxxx\_tdm8<sup>1</sup> beginnt und stelle sicher, daß sie wie folgt aussieht:

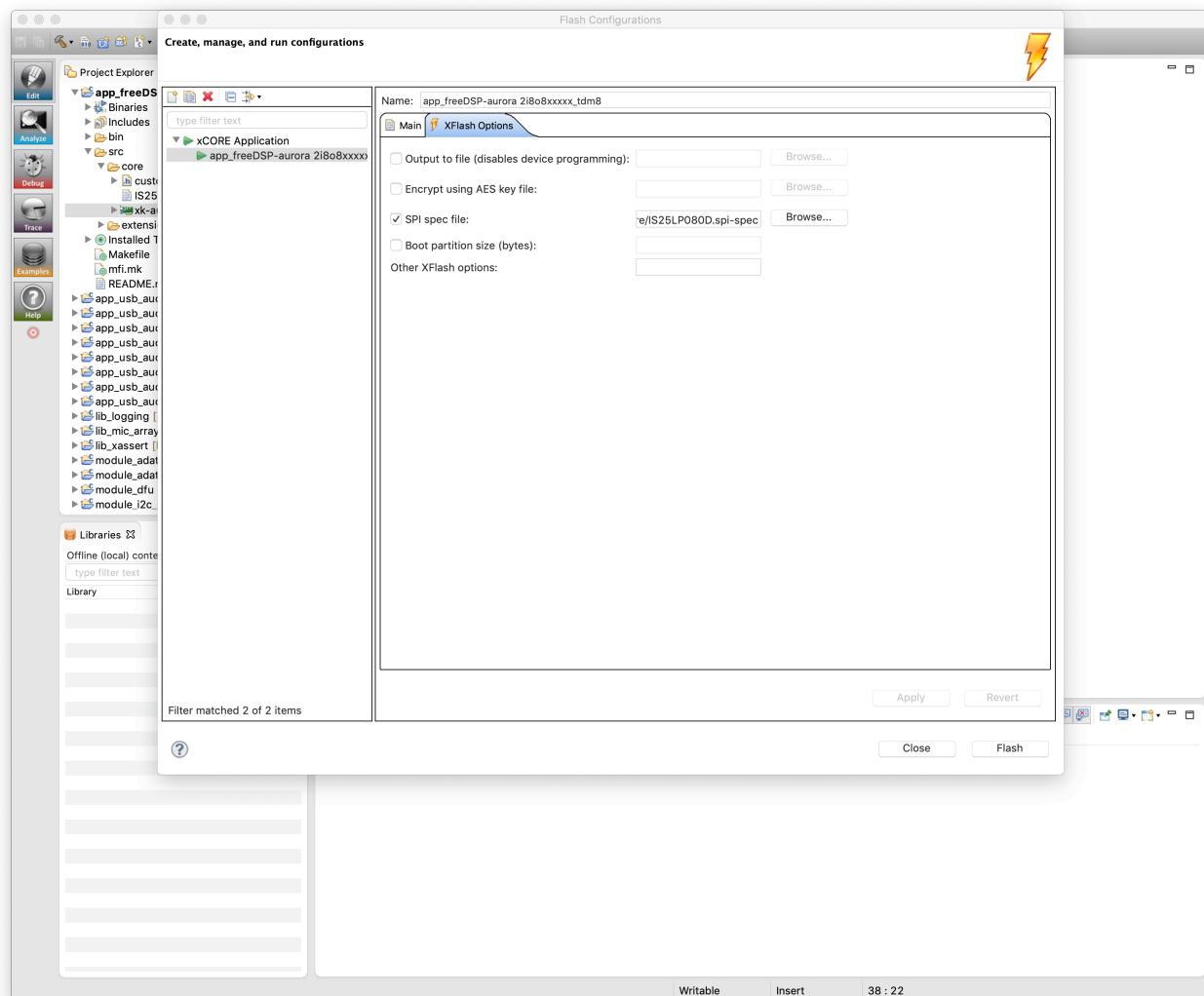
```
XCC_FLAGS_2i808xxxxx_tdm8 = $(BUILD_FLAGS) -DI2S_CHANS_DAC=8 \
-DI2S_CHANS_ADC=8 -DNUM_USB_CHAN_OUT=8 -DNUM_USB_CHAN_IN=8 \
-DMIDI=0 -DSPDIF_TX=0 -DSPDIF_RX=0 -DADAT_TX=0 -DADAT_RX=0 \
-DDSD_CHANS_DAC=0 -DI2S_MODE_TDM=1 -DMIN_FREQ=48000 \
-DMAX_FREQ=48000
```
12. Wähle als aktuelle Build-Konfiguration die aus Schritt 11, z.B. *Project->Build Configurations->Set Active->2i808xxxxx\_tdm8*
13. Klicke nun *Project->Build all*, um die Firmware zu kompilieren.
14. Verbinde Deinen xTAG mit X402 auf dem freeDSP-aurora-Modul.
15. Öffne *Run->Flash Configurations*
16. Wähle xCORE Application und klicke auf die New-Schaltfläche.
17. Vergewissere Dich, daß die Einstellungen wie im folgende Screenshot aussehen.  
Lediglich [Target ID] kann unterschiedlich sein.

---

<sup>1</sup> Du kannst auch eine andere Konfiguration auswählen abhängig von den gewünschten Features, z.B. ADAT



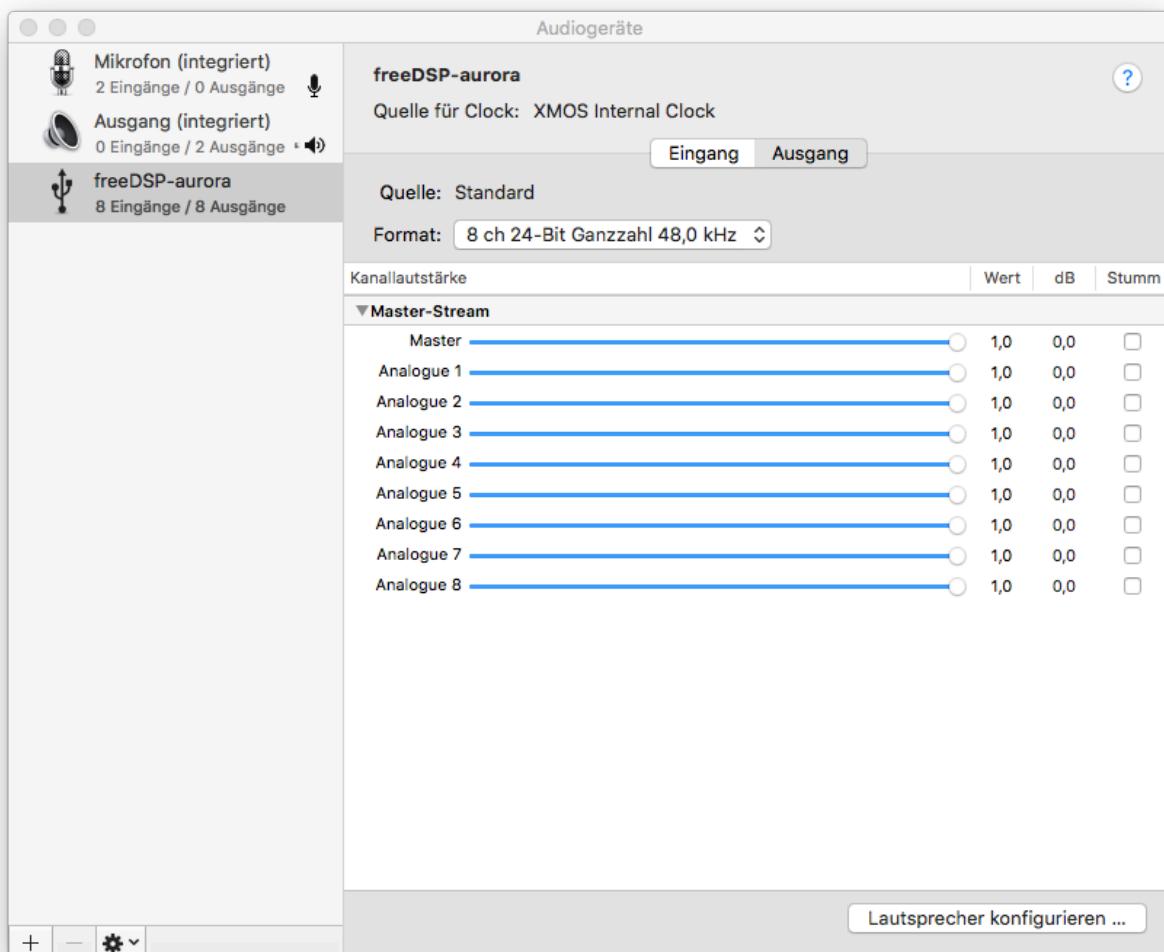
18. Wechsle zu *XFlash Options*, setze den Haken vor SPI spec file und wähle die spi-spec-Datei aus dem freeDSP-aurora-Repository.



19. Klicke auf *Apply*

20. Klicke auf *Flash*, um die Firmware auf Dein Modul hochzuladen.

21. Nach erfolgreichem Abschluss entferne den xTAG-Adapter und verbinde Deinen freeDSP-aurora mit Deinem Computer über den USB-Anschluss. Das Betriebssystem sollte freeDSP-aurora als Audiodevice erkennen und z.B. unter macOS im Audio-MIDI-Setup auflisten:





## HOCHLADEN EINES DSP-PLUGINS

FreeDSP-aurora wurde so entworfen, daß Du verschiedene DSP-Plugins ausführen kannst. Ein DSP-Plugin ist eine Firmware, die auf den SigmaDSP hochgeladen wird. Jedes Plugin definiert eine andere Funktion des DSPs.

In der Regel werden DSP-Plugins mit der App auverdionControl hochgeladen. Im auverdionControl-Manual findest Du dazu weitere Informationen.

Wenn Du auverdionControl auf die neueste Version aktualisierst, hast Du Zugriff auf alle DSP-Plugins, die frei erhältlich sind.

Alternativ kannst Du ein DSP-Plugin auch über einen USBi-Programmer entweder von Analog Devices oder aus dem freeDSP-Projekt hochladen. Über den USBi-Programmer kann SigmaStudio direkt mit dem DSP kommunizieren und Programme hochladen. Dies ist eine komfortable Möglichkeit, um neue Plugins zu erstellen und zu testen.

Verbinde Deinen USBi-Programmer mit X104. Die Vorgehensweise ist wie bei anderen freeDSPs basierend auf ADAU1452. Weitere Informationen findest Du in den entsprechenden Handbüchern oder im EVAL-ADAU1452MINIZ User Guide ([http://www.analog.com/media/en/technical-documentation/user-guides/EVAL-ADAU1452MINIZ\\_User\\_Guide.pdf](http://www.analog.com/media/en/technical-documentation/user-guides/EVAL-ADAU1452MINIZ_User_Guide.pdf)). SigmaStudio kann hier [http://www.analog.com/en/design-center/processors-and-dsp/evaluation-and-development-software/ss\\_sigst\\_02.html](http://www.analog.com/en/design-center/processors-and-dsp/evaluation-and-development-software/ss_sigst_02.html) kostenlos heruntergeladen werden.

Der einfachste Weg, Dein selbsterstelltes DSP-Plugin nichtflüchtig auf freeDSP-aurora zu speichern ist, es zu auverdionControl hinzuzufügen und die GUI entsprechend anzupassen. Der Quelltext von auverdionControl zeigt Dir, wie Du dafür vorgehen musst. Alternativ kannst Du IC102 mit einem 25AA1024T-I/S-M-EEPROM bestücken, über SigmaStudio das EEPROM programmieren und den Jumper JP101 entfernen. Zusätzlich musst Du in der Firmware des ESP32 den DSP-Plugin-Upload entfernen. Dieses Vorgehen erfordert etwas mehr Programmiererfahrung.

## ANHANG

### STÜCKLISTE

Reference	Qty	Value1	Value2	Manufacturer	pn
C103	1	5n60	10% 50V X7R	Yageo	CC0805KRX7R9 BB562
C101, C104, C111, C117, C119, C120, C122, C123, C125, C127, C132, C259, C260, C337, C360, C409, C410, C412, C414, C415, C502, C507, C508	23	10u0	20% 25V X5R	Murata Electronics North America	GRM21BR61E10 6MA73L
C102, C105, C109, C110, C118, C121, C124, C128, C129, C130, C131, C133, C134, C135, C136, C220, C230, C234, C242, C249, C261, C262, C264, C273, C274, C275, C276, C277, C278, C279, C280, C320, C328, C336, C344, C352, C361, C362, C363, C364, C365, C366, C367, C368, C369, C370, C371, C372, C373, C374, C375, C376, C377, C402, C403, C404, C405, C406, C407, C408, C411, C413, C501, C503, C506	65	100n	5% 50V X7R	KEMET	C0805C104K5R ACTU
C106	1	150p	5% 50V C0G	KEMET	C0805C151J5G ACTU
C107, C108	2	22p0	5% 50V C0G/ NPO	KEMET	C0805C220J5G ACTU

Reference	Qty	Value1	Value2	Manufacturer	pn
C112, C113, C114, C137, C138, C203, C204, C211, C212, C217, C218, C219, C227, C228, C229, C233, C237, C238, C241, C247, C248, C255, C256, C269, C270, C301, C302, C309, C310, C314, C315, C321, C324, C325, C329, C330, C331, C340, C341, C345, C346, C347, C353, C356, C357	45	100u	20% 25V	Nichicon	UVY1E101MDD
C115, C116, C126	3	330u	20% 25V	Wurth Electronics Inc.	860010474012
C201, C210, C225, C226, C240, C252, C267, C268	8	1n00	5% 50V PPS	Panasonic Electronic Components	ECH- U1H102JX5
C205, C206, C213, C214, C221, C222, C231, C232, C236, C243, C244, C246, C257, C258, C271, C272, C281, C282, C283, C284, C285, C286, C287, C288	24	100p	5% 50V C0G/ NP0	KEMET	C0805C101J5G ACTU
C202, C207, C208, C209, C215, C216, C223, C224, C235, C239, C245, C251, C253, C254, C265, C266	16	220p	2% 50V PPS	Panasonic Electronic Components	ECH- U1H221GX5
C250	1	4u70	20% 25V X5R	Murata Electronics North America	GRM21BR61E47 5MA12L
C263, C401	2	10n0	10% 50V X7R	Yageo	CC0805KRX7R9 BB103
C305, C306, C318, C319, C334, C335, C350, C351	8	27p0	5% 50V C0G/ NP0	KEMET	C0805C270J5G ACTU
C307, C308, C322, C323, C338, C339, C354, C355	8	3n90	5% 16V PPS	Panasonic Electronic Components	ECH- U1C392JX5

Reference	Qty	Value1	Value2	Manufacturer	pn
C313	1	1u00	10% 16V X7R	Yageo	CC0805KKX7R7 BB105
C303, C304, C311, C312, C316, C317, C326, C327, C332, C333, C342, C343, C348, C349, C358, C359	16	470p	2% 50V PPS	Panasonic Electronic Components	ECH- U1H471GX5
C416, C504	2	22u0	20% 10V X5R	Murata Electronics North America	GRM21BR61A22 6ME51L
C417, C418, C419, C420, C421, C422, C423, C424, C425, C426, C427, C428, C429, C430, C431, C432, C433, C434, C435, C436, C437, C438, C439, C440, C441	25	100n	10% 10V X7R	Murata Electronics North America	GRM155R71A10 4KA01D
D401	1	MBR120VLSFT	Schottky 20V 1 A	ON Semiconducto r	MBR120VLSFT3 G
FER101, FER102	2	600R	@100MHz 600mA	Murata Electronics North America	BLM21AG601SN 1D
IC101	1	ADAU1452	-	Analog Devices Inc.	ADAU1452WBCP Z
IC102	1	DNP	1Mbit	Microchip Technology	25AA1024T-I/ SM
IC103	1	TLE2426ID	Spannungsrefe renz	Texas Instruments	TLE2426IDR
IC104	1	ADM811TARTZ	Spannungsmoni tor	Analog Devices Inc.	ADM811TARTZ- REEL
IC105, IC107	2	LM1117MP-5. 0	LDO	Texas Instruments	LM1117MP-5.0 /NOPB
IC106, IC409, IC502	3	NCP1117LPST 33	LDO	ON Semiconducto r	NCP1117LPST3 3T3G

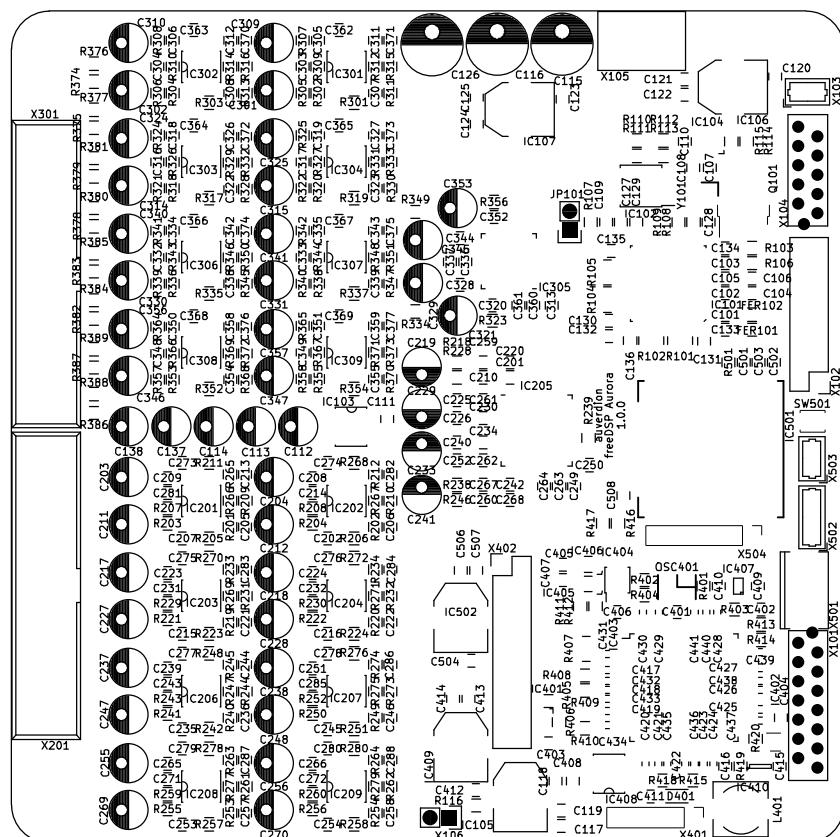
Reference	Qty	Value1	Value2	Manufacturer	pn
IC201, IC202, IC203, IC204, IC206, IC207, IC208, IC209, IC301, IC302, IC303, IC304, IC306, IC307, IC308, IC309	16	OPA1652AIDR	OpAmp	Texas Instruments	OPA1652AIDR
IC205	1	AK5558VN	ADC	AKM Semiconductor Inc.	AK5558VN
IC305	1	AK4458VN	DAC	AKM Semiconductor Inc.	AK4458VN
IC401	1	NC7WZ07	Noninverting Buffer	ON Semiconductor	NC7WZ07P6X
IC402	1	NC7SZ175	D-Type Flip-Flop with Asynchronous Clear	Texas Instruments	SN74LVC1G175DCKR
IC403	1	XE216-512-TQ128	xCore	XMOS	XE216-512-TQ128-C20
IC404	1	CS2100CP	Fractional-N Clock Multiplier	Cirrus Logic Inc.	CS2100CP-CZZ
IC405, IC406	2	NC7SZ157	2-Input Non-Inverting Multiplexer	ON Semiconductor	NC7SZ157P6X
IC407	1	TLV70025	2.5V, 200mA	Texas Instruments	TLV70025DCKR
IC408	1	IS25LP080D	8Mb	ISSI, Integrated Silicon Solution Inc	IS25LP080D-JNLE
IC410	1	TLV62565	1.5A	Texas Instruments	TLV62565DBVR
IC501	1	ESP32-WROOM-32U	Module	Espressif Systems	ESP32-WROOM-32U
JP101	1	Jumper	-	Wurth Electronics Inc.	61300211121
L401	1	2u20	30% 4.5A 26 mOhm	Murata Power Solutions Inc.	45222C

Reference	Qty	Value1	Value2	Manufacturer	pn
OSC401	1	24M0	-	Abracon LLC	ASFL1-24.000 MHZ-EC-T
Q101	1	STD2805	PNP 60V 5A 150MHz 15W	STMicroelectronics	STD2805T4
R101, R102, R104, R105, R407	5	33R0	5% 62,5mW	Yageo	TC164- JR-0733RL
R103, R114, R410	3	1K00	1% 0.125W Thick	Yageo	RC0805FR-071 KL
R106	1	4K32	1% 0.125W Thick	Yageo	RC0805FR-074 K32L
R108	1	100R	1% 0.125W Thick	Yageo	RC0805FR-071 00RL
R109, R239, R402, R404, R408, R409, R411, R412	8	33R0	1% 0.125W Thick	Yageo	RC0805FR-073 3RL
R110, R111, R416, R417	4	2K00	1% 0.125W Thick	Yageo	RC0805FR-072 KL
R107, R112, R113, R401, R405, R406, R413, R414, R501	9	10K0	1% 0.125W Thick	Yageo	RC0805FR-071 0KL
R115, R374, R375, R376, R377, R378, R379, R380, R381, R382, R383, R384, R385, R386, R387, R388, R389, R418	18	100K	1% 0.125W Thick	Yageo	RC0805FR-071 00KL
R116	1	560R	1% 0.125W Thick	Yageo	RC0805FR-075 60RL
R203, R204, R207, R208, R221, R222, R229, R230, R241, R243, R250, R252, R255, R256, R259, R260	16	10K0	1% 0.25W Thin	Stackpole Electronics Inc.	RNCP0805FTD1 0K0
R201, R202, R209, R210, R219, R220, R231, R232, R240, R244, R249, R253, R254, R261, R262, R273	16	6K80	0.5% 0.125W Thin	Panasonic Electronic Components	ERA-6AED682V

Reference	Qty	Value1	Value2	Manufacturer	pn
R205, R206, R211, R223, R224, R242, R248, R251, R257, R258, R268, R270, R272, R276, R278, R280	16	10R0	1% 0.25W Thin	Stackpole Electronics Inc.	RNCP0805FTD1 0R0
R218, R228, R238, R246, R323, R334, R349, R356	8	20R0	1% 0.125W Thick	Yageo	RC0805FR-072 0RL
R212, R233, R234, R245, R247, R263, R264, R265, R266, R267, R269, R271, R274, R275, R277, R279	16	2K20	0.5% 0.125W Thin	Panasonic Electronic Components	ERA-6AED222V
R301, R303, R311, R313, R317, R319, R328, R330, R335, R337, R345, R347, R352, R354, R368, R370	16	4K42	0.5% 0.1W Thin	Susumu	RR1220P-4421 -D-M
R302, R304, R315, R316, R318, R320, R332, R333, R336, R338, R350, R351, R353, R355, R372, R373	16	3K40	0.5% 0.1W Thin	Susumu	RR1220P-3401 -D-M
R305, R306, R312, R314, R321, R322, R329, R331, R339, R340, R346, R348, R357, R358, R369, R371	16	150R	1% 0.25W Thin	Stackpole Electronics Inc.	RNCP0805FTD1 50R
R307, R308, R309, R310, R324, R325, R326, R327, R341, R342, R343, R344, R364, R365, R366, R367	16	4K70	1% 0.125W Thin	Yageo	RT0805FRE074 K7L
R403	1	4R70	1% 0.125W Thick	Yageo	RC0805FR-074 R7L
R415	1	43R2	1% 0.125W Thick	Panasonic Electronic Components	ERJ-6ENF43R2 V
R419	1	80K6	1% 0.125W Thick	Yageo	RC0805FR-078 0K6L
R420	1	120K	1% 0.125W Thick	Yageo	RC0805FR-071 20KL

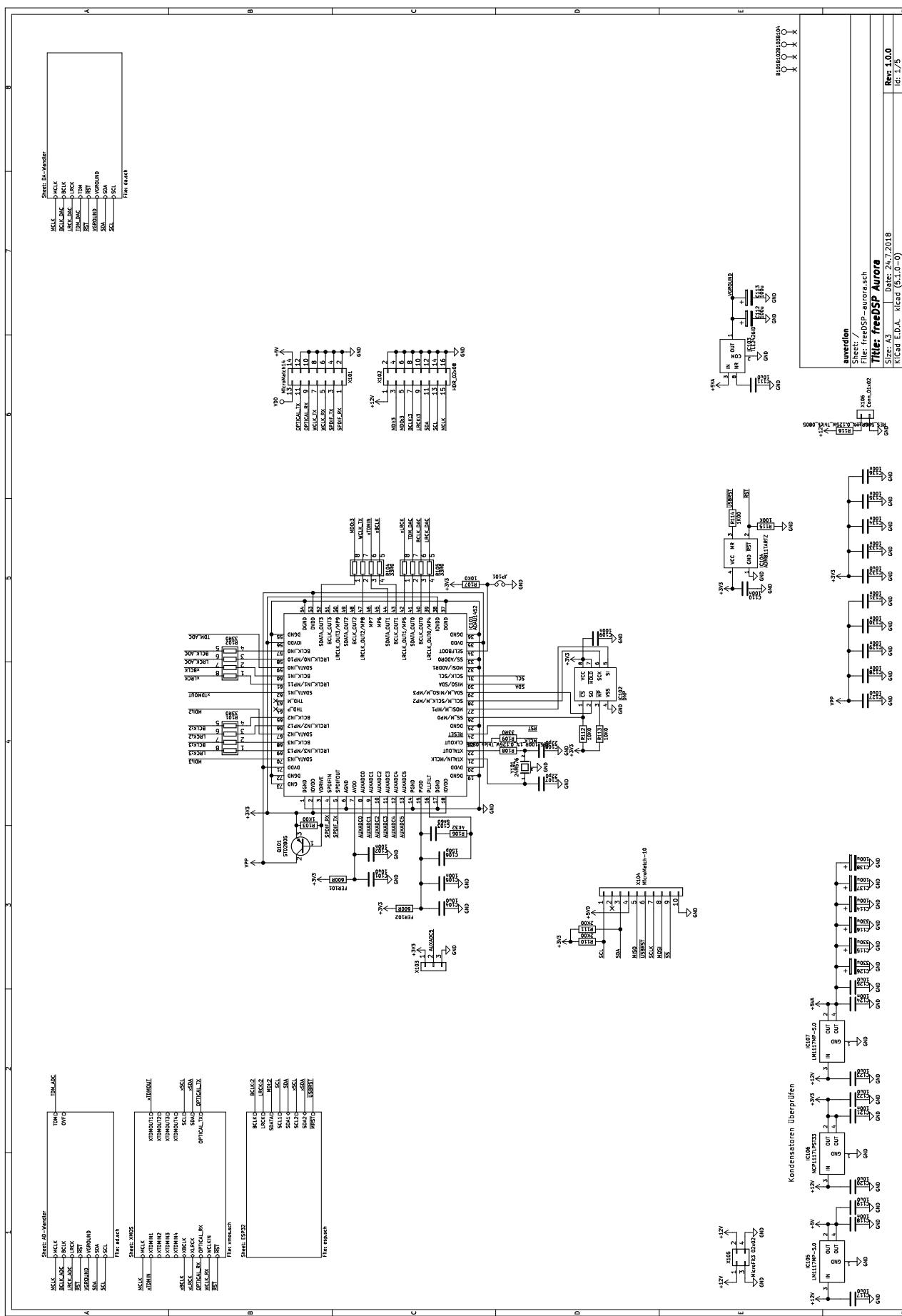
Reference	Qty	Value1	Value2	Manufacturer	pn
SW501	1	B3U-1000P	SWITCH TACTILE SPST- NO 0.05A 12V	Omron Electronics Inc-EMC Div	B3U-1000P
X101	1	MicroMatch-14	-	TE Connectivity AMP Connectors	1-215079-4
X102	1	HDR_02x08	-	Sullins Connector Solutions	PRPC008DAAN-RC
X103, X503	2	PB-01x03	-	Molex, LLC	530470310
X104	1	MicroMatch-10	-	TE Connectivity AMP Connectors	1-215079-0
X105	1	MicroFit3 02x02	-	Molex, LLC	430450428
X106	1	HDR-01x02	-	Wurth Electronics Inc.	61300211121
X201, X301	2	IDC26	-	Bud Industries	BC-32677
X401	1	HDR-01x05	-	Sullins Connector Solutions	PRPC005SAAN-RC
X402	1	HDR-02x10	-	Sullins Connector Solutions	PRPC010DAAN-RC
X501	1	KK254-01x04	-	Molex, LLC	22232041
X502	1	PB-01x05	-	Molex, LLC	530470510
X504	1	HDR-01x06	-	Sullins Connector Solutions	PRPC006SAAN-RC
Y101	1	24M576	-	EPSON	FA-238 24.5760MB-K3
	1	RF ANT 2.4GHZ	WHIP TILT CABLE	PulseLarsen Antennas	W1049B050

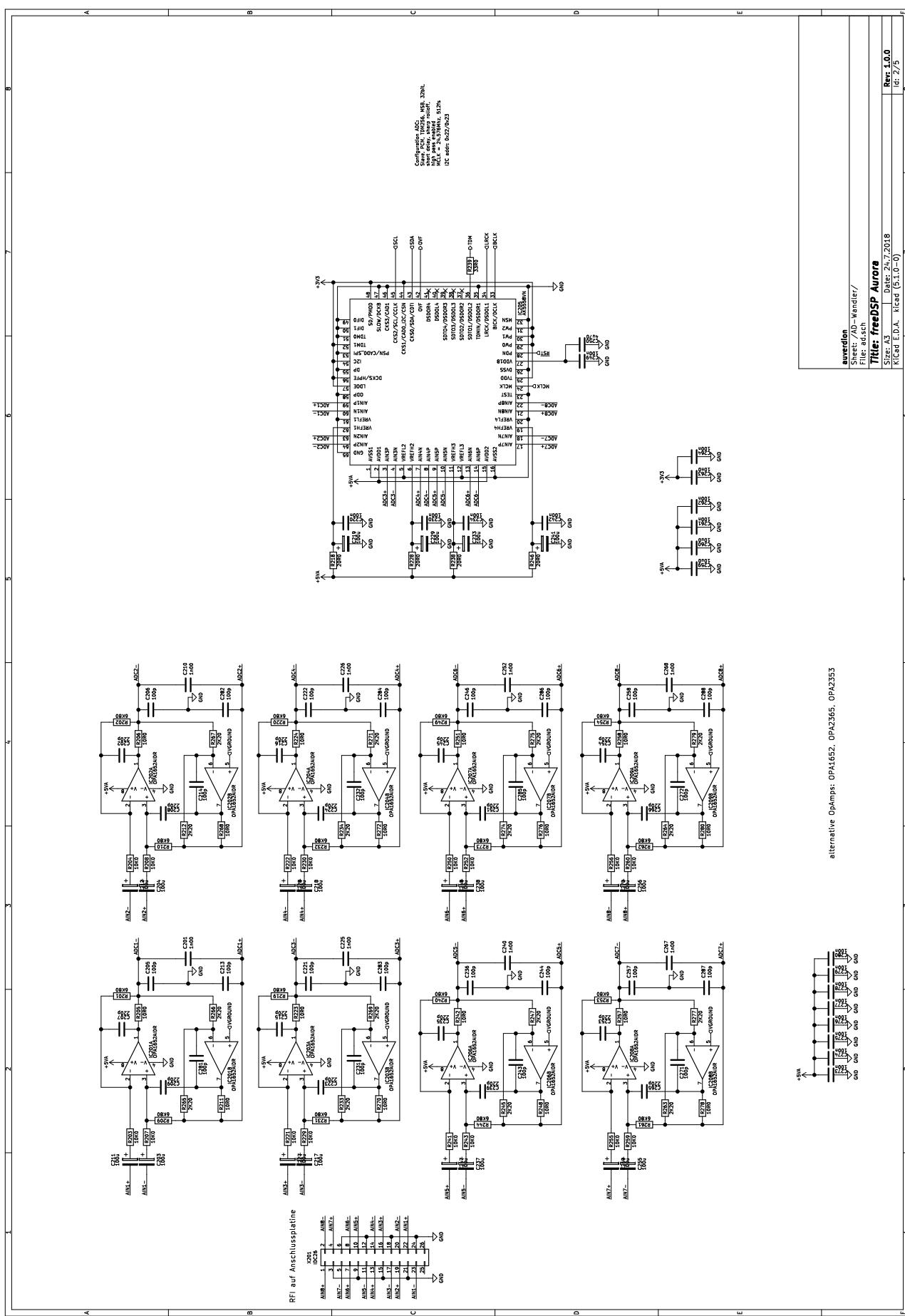
# BESTÜCKUNGSPLAN





## SCHALTPLÄNE





alternative OpAmps: OPA1652, OPA2365, OPA2355

auverdion

Stefan Auverdion

Foto: Stefan Auverdion

Title: **freeDSP Aurora**

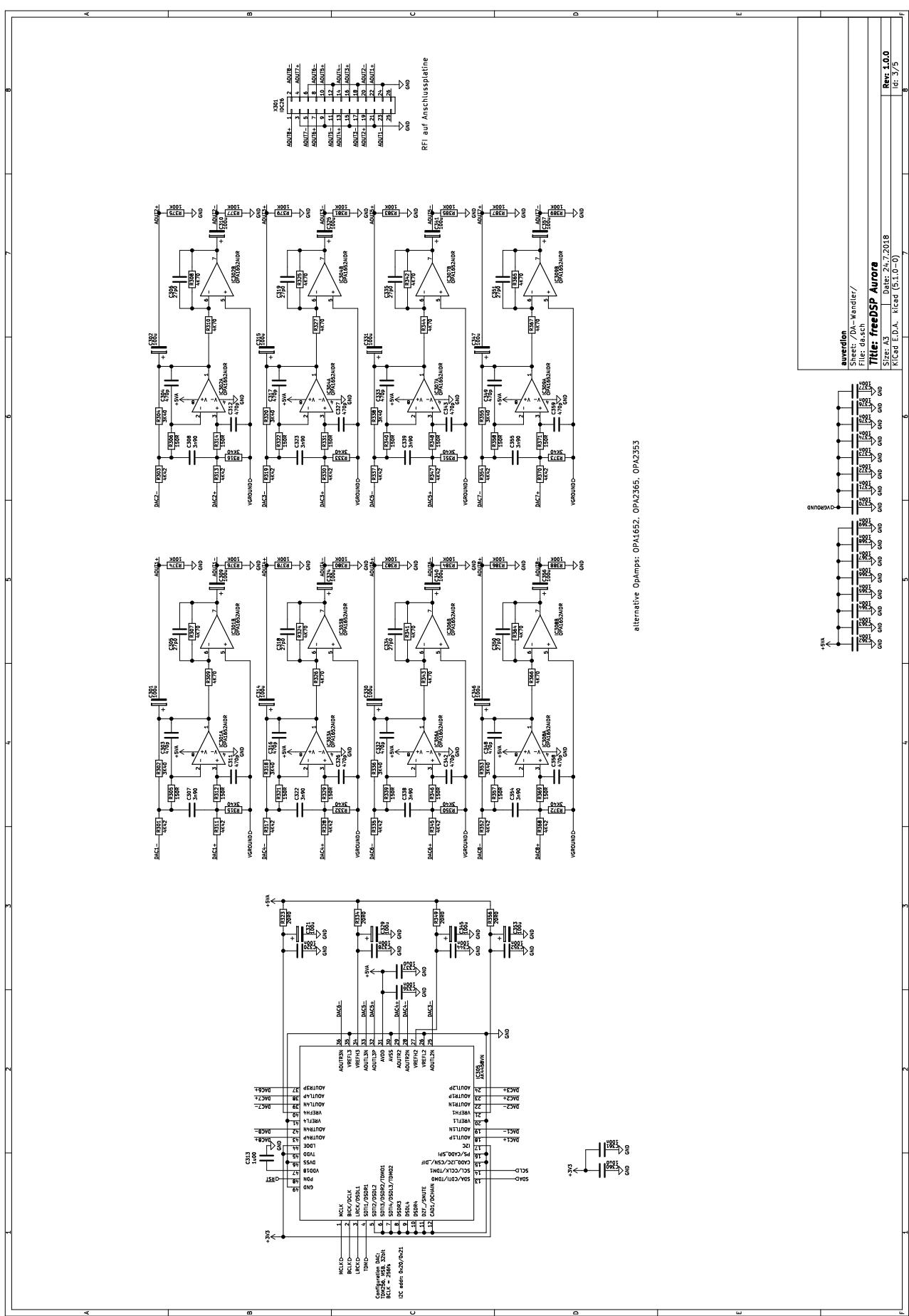
Date: 24.-7.2013

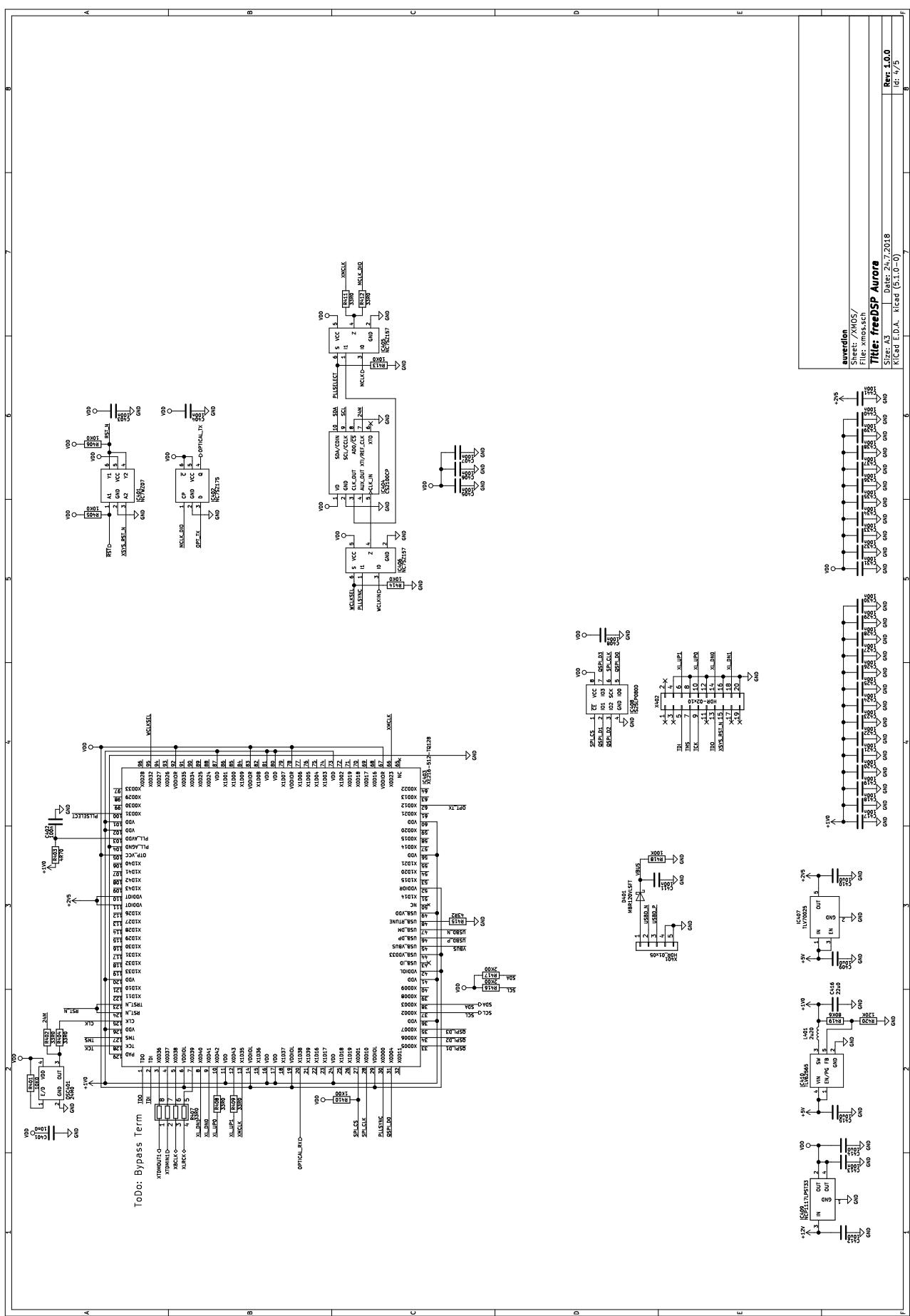
Size: A3

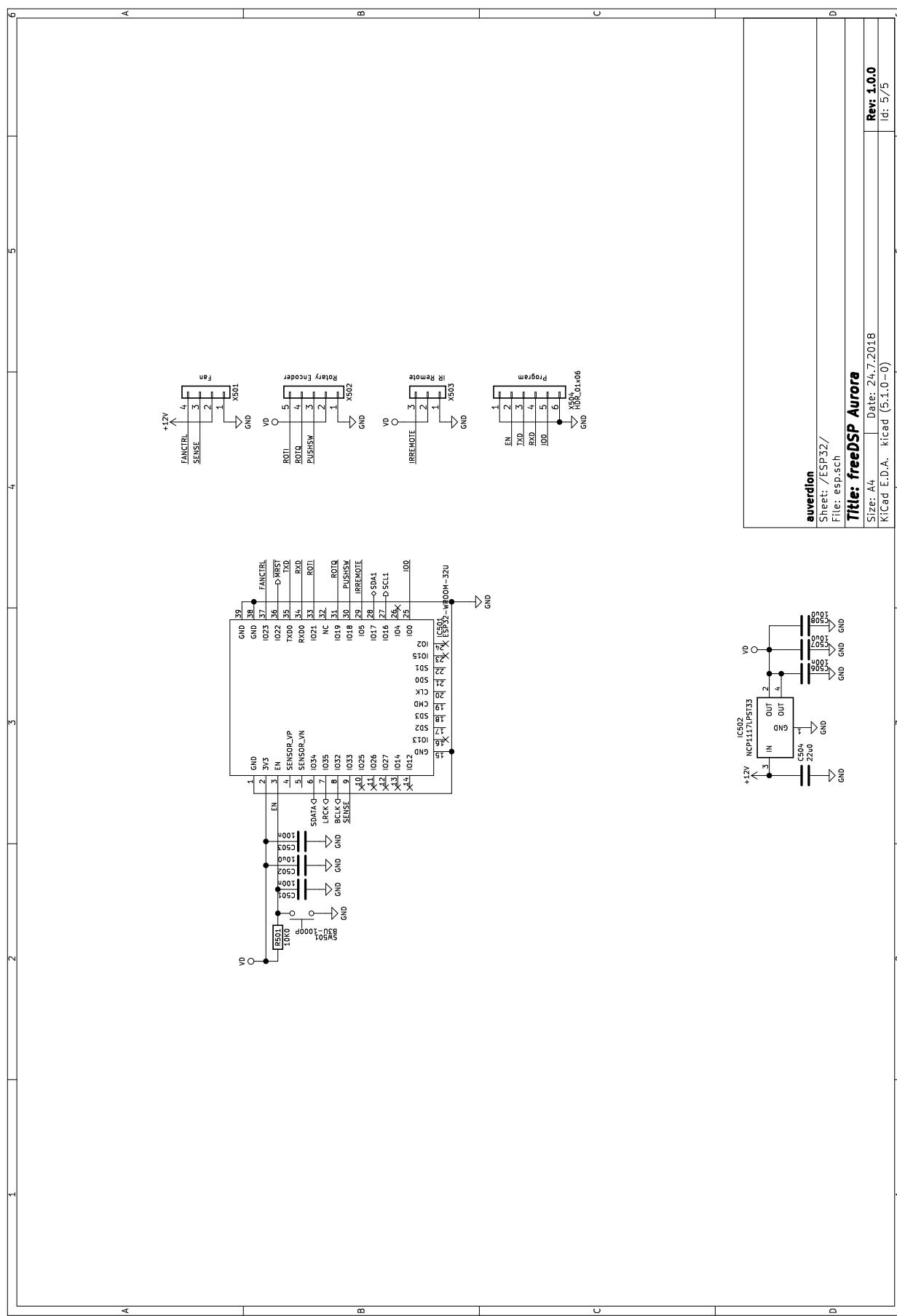
Page: 0.0

Rev: 0.0

Page: 1/5









Bitte besuche <https://github.com/freeDSP/freeDSP-AURORA> für Updates, Bugfixes und neue DSP-Plugins.

Viel Spaß mit freeDSP-aurora und baue etwas besonderes!