
Didactic platform with a DSP to support the teaching of digital signal processing

June 12, 2022

bachelor's thesis submitted by
ANDREI ALEXANDRU TATAR DUMA
in partial fulfillment of the requirements for the
ICT SYSTEMS ENGINEERING DEGREE

Advisor: Francisco Javier Moncunill Geniz

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 3.0 Unported” license.



In glorious dedication to all the teachers that have supported
my way in learning through this undergraduate study.

Abstract

If we put ourselves in the context of an engineering student, we will discover that one of the greatest motivations for learning are laboratory works. This final degree thesis will be about the research and development of a didactic platform for the subject of “Digital Signal Processing”, taught during the third academical year of the ICT Systems Engineering degree. This platform we are going to develop will encase a digital signal processor (DSP) and the required peripherals for the students and teachers to quickly create projects in a fast prototyping environment. Additionally, the provided annexes should meet with the requirements for those who are interested to manufacture our design with little trouble.

Resum

Si ens posem en context d'un estudiant d'enginyeria, descobrirem que una de les majors motivacions de l'aprenentatge són les pràctiques de laboratori. Aquest treball de fi de grau tractarà sobre la recerca i el desenvolupament d'una plataforma didàctica per a l'assignatura de «Processament Digital del Senyal», impartida durant el tercer curs acadèmic del grau d'Enginyeria de Sistemes TIC. Aquesta plataforma que desenvoluparem inclourà un processador de senyals digitals (DSP) i els perifèrics necessaris perquè els estudiants i professors creïn projectes en un entorn de prototipatge ràpid. A més, els annexos proporcionats haurien de complir amb els requisits per a que aquells que estiguin interessats puguin fabricar el nostre disseny amb poques dificultats.

Contents

Abstract	i
Resum	i
List of Figures	v
List of Tables	vii
Acronyms	vii
Units	xi
Author's notes	xiii
I. Report	1
1. Introduction	3
1.1. In context – the subject of digital signal processing at EPSEM	3
1.2. Objectives	3
2. Background	5
2.1. About the subject of digital signal processing	5
2.2. About digital signal processors	5
2.3. Tools and environments used	6
3. Research	7
3.1. The initial design requirements	7
3.1.1. A graphical digital signal processor programming environment	7
3.1.2. The students materials	8
3.1.3. The Audio Phone Connector	9
3.2. Some readily available digital signal processor development kits	10
3.2.1. The ARM University Program Education Kits	11
3.2.2. Microchip Technology DSPic Explorer Development Platform	11
3.2.3. Analog Devices Development Boards	13
3.3. Integrated Development Environments from Analog Devices	14
3.3.1. VisualDSP++	14
3.3.2. CrossCore Embedded Studio	14
3.3.3. SigmaStudio	14
3.3.4. Which is the best for students?	17

3.4.	Choosing a DSP	17
3.4.1.	The SigmaDSP audio processor line	18
3.4.2.	SigmaDSPs with integrated CODECs	19
3.4.3.	SigmaDSPs without integrated CODECs	19
3.4.4.	Choosing a DSP: Conclusion	20
3.5.	The ADAU1467 DSP	20
3.5.1.	The I ² S serial bus	21
3.6.	From Digital to Analog, and Vice Versa	22
3.6.1.	The AD193x Family of CODECs	22
3.6.2.	The AD1939	24
3.6.3.	Single-ended and differential signals	26
3.6.4.	Conversion from balanced to unbalanced	27
3.6.5.	Conversion from unbalanced to balanced	29
3.6.6.	Bypassing the DSP	29
4.	Development	33
4.1.	Our KiCAD Project	33
4.1.1.	KiCad library convention	33
4.1.2.	The PCB guideline we have followed	33
4.2.	Power management	34
4.2.1.	Bypass capacitors	35
4.2.2.	Manual reset and voltage supervision	35
4.3.	Interface with a computer	36
4.3.1.	The adapter for CY7C68013A to Aardvark/USBi programmer	37
4.4.	Selecting the input clock	39
4.4.1.	The ADAU1467 requirements	39
4.4.2.	The AD1937 requirements	39
4.5.	How we routed the CODEC to the DSP	40
4.6.	Characteristics of the DSP Development Board	40
4.7.	How the DSP Development board should look like	41
4.7.1.	3D View	41
4.8.	Schematics of the DSP Development Board	43
5.	Conclusions & Future Work	53
5.1.	Future work	54
Bibliography		55
II.	Appendix	65
A.	Cypress CY7C68013A Voltage Level Translator and In-Circuit Programmer	67
A.1.	Tweaking and flashing the firmware of an USBi programmer to convert the CY7C68013A	67
A.2.	3D Object	72
A.3.	The Schematic	73

List of Figures

3.1. ICT Systems support assembly	8
3.2. A 3-pole phone connector, or TRS plug	9
3.3. STM32F4DISCOVERY kit	12
3.4. Microchip <i>Explorer 16/32</i> Development Platform	12
3.5. ADAU1701 Evaluation Board	15
3.6. The EVAL-21489-EZLITE evaluation board	15
3.7. The two most sold DSPs by <i>MiniDSP</i> in do-it-yourself (DIY) kit format	15
3.8. The VisualDSP++ IDE	16
3.9. CrossCore Embedded Studio screenshot	16
3.10. A SigmaStudio project	16
3.11. Functional block diagram of the ADI programming interface	17
3.12. Functional block diagram of ADAU1467	20
3.13. External to internal audio routing overview of the ADAU1467	21
3.14. I2S timing diagram	22
3.15. Functional block diagram of the AD1939 CODEC	23
3.16. Example of AD1939 in AUX Mode	25
3.17. Thevenin's equivalent of the AD193X DAC outputs at 0 dBFS	26
3.18. Instrumentation amplifier used as a line driver	27
3.19. Two voltage followers in parallel	28
3.20. Two voltage followers in parallel with an error amplifier	28
3.22. A transmission gate with a symmetrical power supply	30
3.23. S/PDIF digital TOSLINK bypass using a multiplexer and a demultiplexer	31
4.1. ADAU1467 symbol with helper labels	34
4.2. AD1937 symbol with helper labels	35
4.3. The breakout board confirmed working	37
4.4. U3D object of the DSP Development Platform	41
4.5. Other views of the DSP Development Platform	42
4.6. DSP Development Platform schematic (page 1)	43
4.7. DSP Development Platform schematic (page 2)	44
4.8. DSP Development Platform schematic (page 3)	45
4.9. DSP Development Platform schematic (page 4)	46
4.10. DSP Development Platform schematic (page 5)	47
4.11. DSP Development Platform schematic (page 6)	48
4.12. DSP Development Platform schematic (page 7)	49
4.13. DSP Development Platform schematic (page 8)	50
4.14. DSP Development Platform schematic (page 9)	51
4.15. DSP Development Platform schematic (page 10)	52
A.1. The CY7C68013A breakout board used in this project	67

A.2. Multiple unknown devices in the windows device manager.	68
A.3. Vendor and product IDs of the breakout board.	68
A.4. Windows startup settings showing to select the 7th option.	69
A.5. Window where the user is prompted to choose the driver.	69
A.6. Warning that only shows when the driver signature enforcement is disabled. . .	70
A.7. The device is now recognized in the device manager with the generic driver. . .	70
A.8. EZ-USB Interface after the EEPROM has been flashed.	70
A.9. Cypress USB Console showing the properties of the USBi drivers.	71
A.10.The device is now recognized in the device manager with the SigmaStudio driver.	71
A.11.Demo in SigmaStudio showing the breakout board functioning.	72
A.12.Schematic with connection to the status LED.	72
A.13.U3D object of the CY7C68013A Voltage Level Translator and ICP Programmer	72
A.14.Schematic of the CY7C68013A Voltage Level Translator and ICP Programmer	73

List of Tables

3.1.	Differences between the two most conflicting phone connector standards	9
3.2.	Balanced TRRS phone connector pinouts	10
3.3.	4.4 mm TRRRS or “Pentaconn” connector	10
3.4.	SigmaDSP audio processors	18
3.5.	Unidirectional serial port channel mapping in SigmaStudio	23
3.6.	Line levels and their approximate nominal levels [100, 101]	27
4.1.	Aardvark/USBi in-circuit programmer header interface	38

Acronyms

4PDT / QPDT	quad-pole double-throw
ADC / A/D	analog-to-digital converter
ADI	Analog Devices
AMD	Advanced Micro Devices
ARM	ARM ⁽¹⁾
ASRC	asynchronous SRC
AUX	auxiliary
AV	audio and video
BGA	ball grid array
CAD	computer-aided design
CCES	CrossCore Embedded Studio
CMOS	complementary MOS
CMRR	common-mode rejection ratio
CODEC	CODEC ⁽²⁾
CPLD	complex programmable logic device
CTAN	Comprehensive TeX Archive Network
CTIA	Cellular Telecommunications and Internet Association
DAC / D/A	digital-to-analog converter
DC	direct current
DFT	discrete Fourier transform
DFU	device firmware update
DIY	do-it-yourself
DMA	direct memory access
DSC	digital signal controller
DSP	digital signal processor ⁽³⁾
DTMF	dual-tone multi-frequency signaling
EDA / ECAD	electronic design automation
EEPROM	electrically erasable programmable read-only memory (ROM)
EPSEM	Manresa School of Engineering (<i>catalan</i> : Escola Politècnica Superior d'Enginyeria de Manresa)
ERC	electrical rules check
FPGA	field-programmable gate array
FTDI	Future Technology Devices International
GPIO	general-purpose inputs/output
GPU	graphics processing unit
GUI	graphical user interface
HD	high-definition
HDK	hardware development kit

¹The company was founded as Advanced RISC Machines

²portmanteau of *coder/decoder*

³In some circumstances within the text, a given initialism or acronym may be used to stand for more than one item. For example, DSP is used for both “digital signal processing” and “digital signal processor”. In all cases, the intended use should be evident from the context.

HPA	headphone amplifier
I/O	inputs and outputs
I²C / I₂C	inter-integrated circuit (⁴)
I²S / I₂S	inter-IC sound (⁵)
IC	integrated circuit
ICP	In-Circuit Programmer
ICT	information and communications technology
IDE	integrated development system
JEITA	Japan Electronics and Information Technology Industries Association
KLC	KiCad library convention
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LCD	liquid-crystal display
LDO	low drop-out
LED	light-emitting diode
LFSR	linear feedback shift register
MAC	multiplier and accumulator
MCLK	master clock
MCU	microcontroller unit
MOS	metal-oxide-semiconductor
MSB	most significant bit
MUX	multiplexer
NI	National Instruments
NXP	Next eXPerience Semiconductors (⁶)
OCW	OpenCourseWare
op-amp	operational amplifier
OS	operating system
PCB	printed circuit board
PDM	pulse-density modulation
PDS	digital signal processing (<i>catalan</i> : processament digital del senyal, ⁷)
PLL	phase-locked loop
PWM	pulse-width modulation
RAM	random access memory
RF	radio frequency
RGB	red-green-blue
RMS	root mean square
ROM	read-only memory
S/PDIF	Sony/Philips Digital Interface
SDATA	serial data
SDK	software development kit
SEC	electric control systems (<i>catalan</i> : sistemes electrònics de control)
SHARC	Super Harvard Architecture Single-Chip Computer
SoC	system on a chip
SPDT	single-pole double-throw
SPDT	single-pole single-throw
SPI	serial peripheral interface
SRC	sample rate converter
TDM	time division multiplexion
TFT	thin-film-transistor

⁴Pronounced “eye-squared-C”.

⁵Pronounced “eye-squared-ess”.

⁶Formerly Philips Semiconductors. Not to be confused with the manufacturer Nexpria [1].

⁷ICT Systems subject taught at the EPSEM

TI	Texas Instruments
TOSLINK	Toshiba Link
TRS / TRRS /	
TRRRS / TS	<i>T</i> stands for “tip”, <i>R</i> stands for “ring” and <i>S</i> stands for “sleeve”
TV	television
UPC	Technical University of Catalonia (<i>catalan</i> : Universitat Politècnica de Catalunya, ⁸)
USART	universal synchronous and asynchronous receiver-transmitter
USB	universal serial bus
UVLO	undervoltage-lockout
VFBGA	very thin profile fine pitch ball grid array (BGA)
VSCode	Visual Studio Code
WLCS	waffer-level chip-scale package
WYSIWYG	“what you see is what you get”

⁸Currently referred to as BarcelonaTech

Units

dBV	dBm full scale
dBFS	dBm relative to full scale ⁽⁹⁾
dBm	decibel-milliwatts ⁽¹⁰⁾
dBu	dBm unloaded
% THD+N	% THD + noise
% THD	total harmonic distortion
" / in	inch
Vpk	volts peak
Vpp	volts peak-to-peak
VRMS	volts RMS
IPS	instructions per second

Author's notes

If you are reading the online version of this document please read the following.

This document was compiled using pdfTeX without backreferences enabled, due to various conflicts with different packages. To improve the readers user experience, we recommend using a PDF reader that has keybinds to navigate back and forth through different view. Adobe Reader's keybinds are **[Alt]+[Left/Right Arrow Key]**. This is useful for navigating to a reference and then return back to where the reference was clicked.

Optionally, use a PDF reader compliant with version 1.7 and above because we embedded 3D models in this document.

⁹Assigned to the maximum possible digital level.

¹⁰The power dissipated as heat by a 600Ω resistor.

Part I.

Report

1. Introduction

1.1. In context – the subject of digital signal processing at EPSEM

As part of the OpenCourseWare (OCW) movement, which the Technical University of Catalonia (UPC) is also part of [2], the general public can access the free and open collection of material of the ICT Systems Engineering course imparted at EPSEM¹ [3].

The subject we are focusing on in this thesis is called digital signal processing (PDS). This subject is taught during the first part of the third academical year. You will find a brief description on section 2.1 on page 5. This subject contains in total 5 laboratory practices which are expected to fit in a four-month period, which are:

1. *Pseudorandom Noise Generator* based on a linear feedback shift register (LFSR).
2. *Moving average filter* using the Z-transform.
3. *Digital oscillator* using the Z-transform.
4. *Spectrum of a PWM signal* using the discrete Fourier transform (DFT).
5. *Filtering with a field-programmable gate array (FPGA) chip.*

1.2. Objectives

Despite the variety of laboratory practices presented by the subject of PDS, this final degree thesis will propose a new one from a more familiar perspective for students.

Given the strategy of first learning from the roots, it has been found relevant that the subject of PDS should not only deal with the construction of algorithms on Octave or Matlab, Arduino or FPGAs, but should also allow the student to work from a higher level, *i.e.* as if it was working with a high-level programming language, such as Python.

In summary, the main objective of this final degree project is to design a platform on which to develop DSP “circuits” imitating high-level programming. To accomplish this, we will need to:

1. Research into what is the essence that makes a good DSP Development Platform, *i.e.*, the available on-chip resources, the flexibility of the platform, the programming environment required, ...
2. Out of the researched information, we will set our own requirements for the development platform to be successful.
3. Then, we will make a schematic with our design decisions.
4. After that, we will elaborate a final PCB ready to be manufactured.
5. And for the closing, a set of conclusions will summarize the main achievements of the thesis and present future work.

¹Manresa School of Engineering

2. Background

In this chapter, we will name some of the previous works that have been done related with the theme of this final degree project. Also, the tools, languages and environments used for project development.

2.1. About the subject of digital signal processing

“ The course is a continuation of, and follows a script similar to that of, *Signals and Systems* dedicated to analog signal processing. During the course, digital tools are presented and compared with the analog tools as seen previously: Z-transform vs Laplace transform, Fourier transform vs discrete Fourier transform, analog filters vs digital filters... The processing will be simulated in a computer using high-level programming languages (Octave or Matlab) and basic (real-time) applications will be implemented on a microcontroller and/or CPLD/FPGA using low-level programming languages (Assembly, C...) In the laboratory we will start from analog signals, convert them into digital ones, simulate or implement their processing, and then turn them back into analog ones. Therefore, the concepts of A/D and D/A converters and the concepts of sampling, quantification and interpolation will need to be introduced. ”

Extracted from OpenCourseWare iTIC [4].

2.2. About digital signal processors

A DSP is a specialized microprocessor chip, with its architecture optimized for the operational needs of digital signal processing, *i.e.* analyzing, modifying, and synthesizing signals such as sound, images, and scientific measurements [5, 6]. DSPs are widely used in audio signal processing, telecommunications, computer vision, speech recognition, and in common consumer electronic devices such as mobile phones, HD TV products and car audio.

One could say that DSPs are analogous to graphics processing units (GPUs) in that the task a general-purpose microprocessor can also execute has been optimized to a separate device that can keep up with it. For GPUs it is parallel processing, and for DSPs it is keeping up with real time computations [7].

Most DSPs use fixed-point arithmetic, because in real world signal processing the additional range provided by floating point is not needed, and there is a large speed benefit and cost benefit due to reduced hardware complexity.

Floating point DSPs may be invaluable in applications where a wide dynamic range is required. Product developers might also use floating point DSPs to reduce the cost and complexity of software development in exchange for more expensive hardware, since it is generally easier to implement algorithms in floating point. Generally, DSPs are dedicated integrated circuits; however DSP functionality can also be produced by using FPGAs [8].

The major manufacturers in the DSP industry are: Texas Instruments (TI) [9], Analog Devices (ADI) [10], Next eXPerience Semiconductors (NXP) [11], ARM [12], Altera (acquired by Intel) [13], Xilinx (acquired by AMD [14]) [15]. Other notable manufacturers are Qualcomm [16], XMOS [17], Cirrus [18], and Microchip [19], known for the AVR family of microcontrollers — originally developed by Atmel, now a subsidiary of Microchip — especially common in hobbyist and educational embedded applications, popularized by their inclusion in many of the Arduino line of open hardware development boards [20].

2.3. Tools and environments used

Many tools were used during the development of this thesis. The list of tools are provided below.

- *Windows 11*, all aspects of this project have been done on this operating system (OS).
- *Visual Studio Code (VSCode)*, one of the most versatile and powerful integrated development systems (IDEs) [21].
- *LATEX workshop*, an extension for VSCode that has been used to compile this document with [22].
- *TEX Live*, a cross-platform, free software distribution for the T_EX typesetting system [23].
- *KiCad 6.0.4*, an Open Source electronic design automation (EDA or ECAD) editor been used to design the printed circuit boards (PCBs) of the device [24].
- *Ultra Librarian*, a free online PCB computer-aided design (CAD) library [25].
- *Mouser Electronics*, a global distributor of semiconductors and electronic components. We used it to find and compare components [26]. Some 3D models were used from this site too.
- *LTSpice XVII*, a high performance SPICE simulator software with a graphical user interface (GUI) schematic editor [27].
- *FreeCAD 0.19.4*, in conjunction with KiCad, this Open Source CAD software has been used to render and export the PCB models as `wrl` files [28].
- *MeshLab*, the open source system for processing and editing 3D triangular meshes [29]. The purpose of this tool was to convert the `wrl` models from KiCAD and to convert them to `u3d` to be able to attach them to this PDF 1.7 document with the LATEX package *media9* [30].

3. Research

3.1. The initial design requirements

For the goal of this thesis, we have to set a baseline in which to center our design decisions henceforward.

3.1.1. A graphical digital signal processor programming environment

The main requirement which has proven to be successful with other didactic platforms is to provide an easy-to-use hardware and software design, which coincides with Arduino's company culture [31].

Arduino is an open-source hardware and software company which manufactures single-board microcontrollers. The microcontroller units (MCUs) can be programmed using C or C++, but the standard library included with *Arduino IDE* software development kit (SDK) make it easy for newcomers to learn to use the platform with any of its hardware development kits (HDKs). Makers alike appreciate all the support they are given, both by the manufacturer and the large community that got involved around this platform.

The students of ICT Systems will eventually see in the subject of electric control systems a piece of software called *The MathWorks Simulink* [32] together with *MATLAB* to combine the power of textual and graphical programming in one environment.

It is not out of the question to use graphical programming IDEs to program DSPs [33]. One such programming environment is National Instruments (NI) Laboratory Virtual Instrument Engineering Workbench (LabVIEW) [34]. This IDE is commonly used for data acquisition, instrument control, test automation, industrial control, embedded system design analysis and signal processing. A similar graphical programming environment is called *FlowStone* by DSP Robotics [35]. This accessible looking software is more geared towards education, sound and music, robotics, home automation, image and PDS.

We would like to mention the FPGA development and education board “DE0-Nano” [36] which our university has used in some courses. This board has an Intel Cyclone IV FPGA which has 22000 logic elements, 4 phase-locked loops (PLLs), 594 Kb, 66 digital signal processor blocks in multiplier and accumulator (MAC) format [37]. Together with the *Quartus Prime Development Suite*, we can develop many applications, *e.g.* a dual-tone multi-frequency signaling (DTMF) decoder using the Goertzel algorithm for computing DFT coefficients and signal spectra using a digital filtering method, which the ICT System students will do during a laboratory practice in the subject of *Embedded Systems* [38, 6].

Thus, we could say that one of our requirements should be to pick only the manufacturers seen in the section 2.2 on page 5 that offer a *graphical programming environment*.

3.1.2. The students materials

The students of ICT Systems at EPSEM are provided with a list of laboratory supplies which they will need throughout the academic degree [39]. These are required to be able to assist to the practical laboratory lessons, which are done in groups of two people. The materials in question are:

- At least one protoboard.
- Seven 4 mm female banana connectors.
- Two female RG-58 BNC connectors.
- Two male-to-male RG-58 BNC cables.
- Two male RG-58 BNC to male 4 mm banana cables.
- Three pairs of male-to-male 4 mm banana cables.
- A small screwdriver, cutting pliers, and optionally, a wire stripper.

All female connectors are for panel mounting. All cables should be at least of 80 cm.

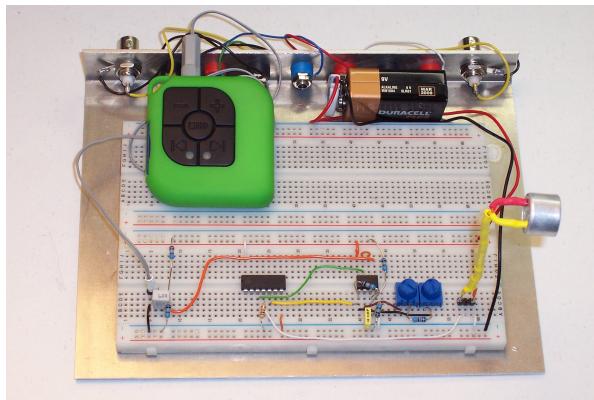


Figure 3.1.: ICT Systems support assembly.

As we can see in the figure above, the materials focus on the use of banana and BNC cables. BNC cables are not commonly used within the audio industry unless they are required for electronic test instruments, *e.g.* oscilloscopes or audio analyzers [40]. On the other hand, banana connectors are more common in laboratory bench equipment, *e.g.* power supplies or waveform generators, but are also used in audio equipment like modular synthesizers [41].

However, we have seen that prototyping environments and development boards use 2,54 mm pin headers and sockets, which are easy to connect to a breadboard with male-to-male or male-to-female jumper cables. Female-to-female jumper cables are also used to connect from board-to-board.

What the users of the development board we are designing will most likely be using, are phone connectors from their laptops or phones. In the next section we will explain the different types of phone connectors that exists.

3.1.3. The Audio Phone Connector

The audio phone connector is cylindrical in shape¹, the male connector having a grooved tip and the female socket having a detent for retaining the male end when inserted. They metallic contacts are called poles. Phone connectors come with at least two poles, but other variants have as many as five.

The most common standard, the RC-5325A [42] published by the Japan Electronics and Information Technology Industries Association (JEITA) [43] in the year 1993, has given a name to each pole:

T: tip.

R: ring, to all the poles in between the tip and sleeve.

S: sleeve, to the pole closest to the cable, and usually featuring a large surface area.

So, a phone connector is called “TS plug” when having 2 poles, “TRS plug” when having 3 poles, “TRRS plug” when having 4 poles, and “Trrrs plug” when having 5 poles. We can see the male TRS plug in the figure below



Figure 3.2.: A 3-pole phone connector, or TRS plug

The TRRS plug has been used in the past to transmit composite video plus two audio channels. Nowadays, the phone connectors are mainly used for sending and receiving audio signals. The fourth pole, or the sleeve when different number of poles is used, is usually used for ground reference. We said “usually” because a wiring convention conflict began when two standards had arose because microphones were introduced to TRRS plugs. The two wiring conventions are shown in the table below. Today, the CTIA [44] standard prevails in most devices.

Table 3.1: Differences between the two most conflicting phone connector standards

Standard	Tip	Ring 1	Ring 2	Sleeve
CTIA	Left Audio Channel	Right Audio Channel	Ground	Microphone
OMTP	Left Audio Channel	Right Audio Channel	Microphone	Ground

Note: The rings are numbered in ascending order starting from the tip.

Phone connectors have different diameters. In a grand majority of devices, the most common diameter used is of 3,5 mm, called “mini plug” for being miniature in size, but the following sizes also exist:

- 2,5 mm, called “sub-mini” for being subminiature in size.
- 4,4 mm, known as the “Bantam” or “TT” connector — audiophiles refer to it as “Pentaconn” [45, 46].
- 6,35 mm, a descendant of the early telephone switchboards and most notably used by professional audio equipment like guitars and amplifiers.

¹Not to be confused with the telephone connector.

Phone connectors can also be wired to allow differential signaling. Stereo differential signaling requires two pairs of wires, making the TRRS jack suitable. High fidelity audio devices use phone connectors over other balanced connectors, *e.g.* the industry standard XLR connector. This is because in small devices XLR connectors do not fit the constraints. The table below show various pinouts of balanced TRRS plugs used by different products.

Table 3.2: 2,5 mm and 3,5 mm TRRS phone connector pinouts [47]

Device	Size	Tip	Ring 1	Ring 2	Sleeve
AK DAP	2,5 mm	R-	R+	L+	L-
Fiiio DAP	2,5 mm	R-	R+	L+	L-
Geek Wave	3,5 mm	L+	R+	L-	R-
HifiMan HM-801	3,5 mm	L+	R+	L-	R-
Oppo PM-3	3,5 mm	L+	R+	L-	R-
Sony MDR-1A	3,5 mm	L+	R+	R-	L-

Audio channels: Left is *L*, and right is *R*.

Polarity: The non-inverting input is + and the inverting input is -, like with operational amplifiers (op-amps).

The 4,4 mm TRRRS phone connector can also fit balanced signals through the pinout shown in the table below, described by the JEITA standard RC-8141C [48].

Table 3.3: 4,4 mm TRRRS or “Pentaconn” connector

Standard	Tip	Ring 1	Ring 2	Ring 3	Sleeve
Pentaconn	Left +	Left -	Right +	Right -	Ground

The Audio Phone Connector: Conclusion

We have chosen to use the 3,5 mm phone connector for the design of the development board because it is a highly versatile and readily available connector. Moreover, the small footprint will save space if over other types of connectors like BNC, as seen in the laboratory equipment of the students in the previous subsection, or RCA connectors, used in analog audio and video (AV) equipment and sometimes by the digital Sony/Philips Digital Interface (S/PDIF) [49].

3.2. Some readily available digital signal processor development kits

This section would like to present how DSP HDK are like. For starters, let us differentiate between some options the section 2.2 on page 5 offer.

3.2.1. The ARM University Program Education Kits

“The ARM University Program has developed a suite of Education Kits in a range of core subjects relevant to Electrical, Electronic and Computer Engineering, Computer Science and beyond. An Education Kit comprises a full set of teaching materials including lecture slides and lab manuals with solutions.” [50] All its teaching materials are published on GitHub [51].

The ARM digital signal processor Education Kit was originally meant to be used with the STM32F4DISCOVERY kit, but it also supports the newer Cortex-M7 based STM32F7 Discovery board [52, 53]. See the figure on the following page. The first board is similar to an Arduino UNO because it serves like a general purpose HDK with an ARM MCU instead of an Atmel AVR, and the only form of inputs and outputs (I/O) are the mini universal serial bus (USB) port, the general-purpose inputs/output (GPIO) pins, two buttons, an red-green-blue (RGB) light-emitting diode (LED) and a headphone jack.

A similar Audio DSP was developed by Skyler Cornell for his 2021 thesis at Bryn Mawr College [55]. His design incorporates a CS5343 ADC [56] and a CS4344 DAC [57] by Cirrus Logic and the STM32F407VGT60 MCU.

The newer Discovery kit with STM32F746NG MCU has ARDUINO connectivity support which provides unlimited expansion capabilities with a large choice of specialized add-on boards, apart from having greater connectivity and even including a 4,3 " color thin-film-transistor (TFT) liquid-crystal display (LCD) with capacitive touch [53].

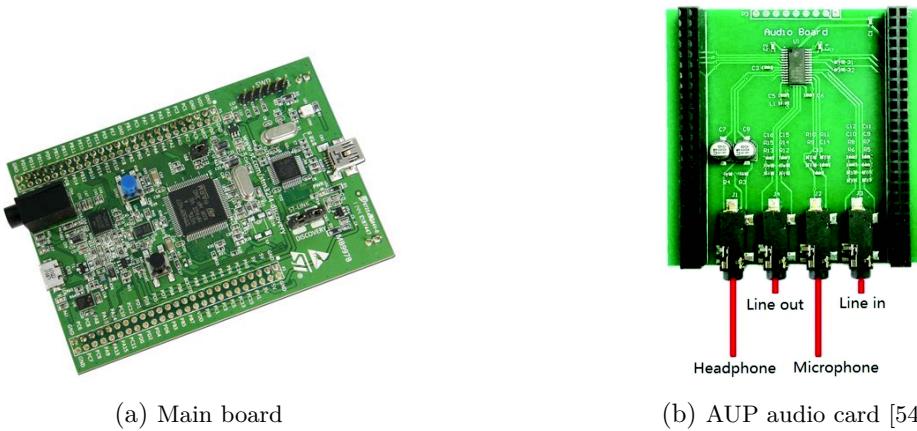
3.2.2. Microchip Technology DSPic Explorer Development Platform

In the catalog of Microchip Technology we find digital signal controllers (DSCs), a type of hybrid MCUs and DSPs. Like microcontrollers, DSCs have fast interrupt responses, offer control-oriented peripherals like PWM and watchdog timers, and are usually programmed using the C programming language, although they can be programmed using the device’s native assembly language. On the DSP side, they incorporate features found on most DSPs such as single-cycle MAC units, barrel shifters, and large accumulators [58].

One such device is the *dsPIC* family of DSCs. The “*dsPIC33* DSCs add digital signal processor (DSP) performance for embedded applications requiring time-critical response while offering the simplicity of an MCU. Offering single-cycle execution, deterministic interrupt response, zero overhead looping and fast direct memory access (DMA), this family also adds a single-cycle 16×16 MAC and 40-bit accumulators, making it ideal for math-intensive, high-performance and robust motor control, digital power, functional safety and security designs.” [19]

Microchip also offers a wide variety of libraries for audio encoding and decoding, USB Audio playback from a personal computer or mobile device, and Bluetooth audio, voice, and data streaming [59].

Amongst its development boards, the *DM240001-3*, also referred as the “EXPLORER 16/32 DEVELOPMENT KIT”, is a modular HDK that permits to swap the MCU through *Processor Plug-In Modules*, add expansion boards and daughter cards from a variety of manufacturers like *MikroElektronika*’s *mikroBUS* [60, 61]. See the figure on the next page



(a) Main board (b) AUP audio card [54]

Figure 3.3.: The STM32F4DISCOVERY kit and its DSP expansion card

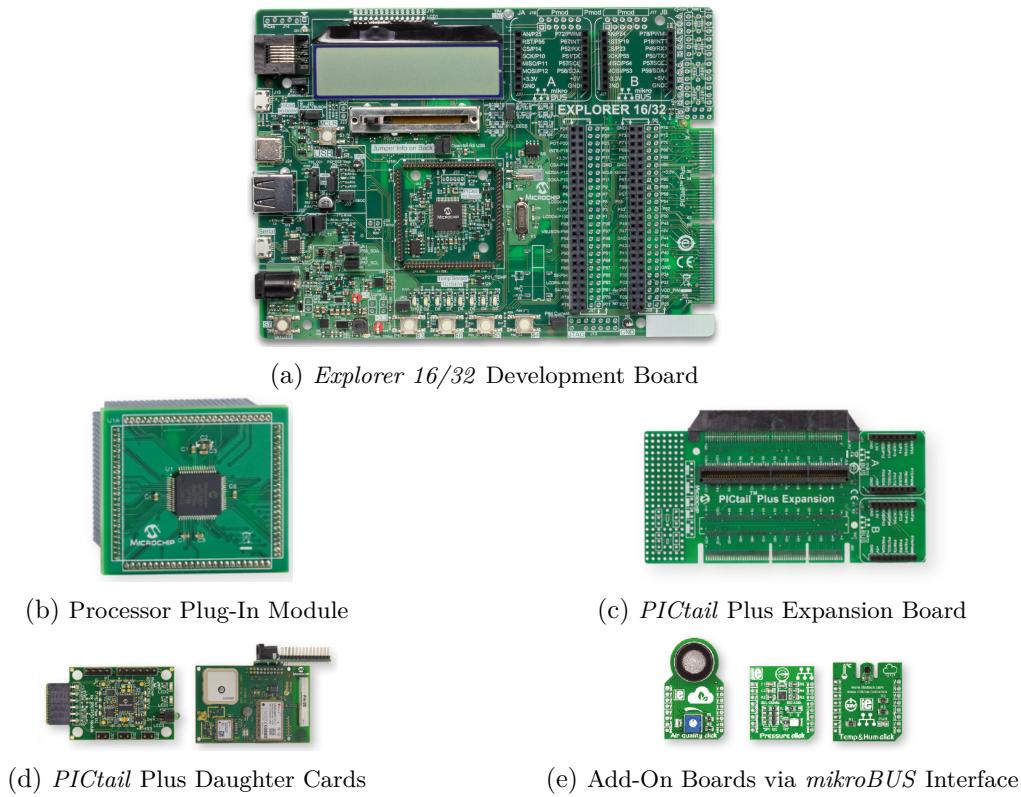


Figure 3.4.: Microchip *Explorer 16/32* Development Platform

3.2.3. Analog Devices Development Boards

Analog Devices products include mixed-signal and digital signal processing ICs. These technologies include data converters, amplifiers, radio frequency (RF) technologies, embedded processors or DSPs, ICs, power management, and interface products.

Working with ADI has some benefits. Its support community called *EngineerZone* makes it easy to make questions and to get answers from professionals in the field [62]. ADI also hosts an education library and academic community [63], and a design center with evaluation HDKs and SDKs to help determine whether a component will fit within our design [64].

From the product catalog of ADI we can find within its *audio and video products* the following families of processors: [65]

- *Blackfin* Embedded Processors are “...16/32-bit embedded processors offer software flexibility and scalability for convergent applications: multiformat audio, video, voice and image processing, multimode baseband and packet processing, control processing, and real-time security.”
- *SHARC* [66] are “...deterministic and very low processing latency with best-in-class MIPS/mW performance. The SHARC processor family dominates the floating-point PDS market with exceptional core and memory performance and outstanding I/O throughput. With multiple product variants and price points, SHARC brings real-time floating-point processing performance to many applications where dynamic range is key.”
- *SigmaDSP* Audio Processors are “...fully programmable, single chip audio DSPs that are easily configurable through the SigmaStudio graphical development tool, and are ideal for automotive and portable audio products. SigmaDSP chips are available with integrated sample rate converters, A/D converters, D/A converters, and output amplifiers.”

SigmaDSP development boards

ADI has many evaluation boards for almost every product they produce. In the PDS department, the *EVAL-ADAU1701* evaluation board is a good implementation example [67]. See the figure on page 15.

This device is based on the *ADAU1701 SigmaDSP 28/56-bit audio processor* which has two ADCs and four DACs, therefore, it also acts as a coder-decoder (CODEC). The I/O of this board consists of:

- Three push-buttons,
- LED,
- a potentiometer,
- two 3,5 mm phone jacks (one for input and the other for the output),
- and also, a low-power Class-D amplifier that can drive a load of 4Ω at 2 W or 8Ω at 1,4 W.

Another fairly common DSP solution between audio enthusiasts is the *MiniDSP 2x4* by MiniDSP. This DSP is based on the same audio processor like the aforementioned evaluation board, but only has 2 RCA inputs and 4 RCA outputs. It comes in two form factors: with a metal enclosure, for consumer use, or without one, for do-it-yourself (DIY) audio applications [69, 70].

SHARC development boards

On the other hand, SHARC systems on chips (SoCs) do not have CODECs built-in, so they are dependent on the external peripherals they may have connected. Since they have more capabilities than SigmaDSP evaluation boards, the expansion options they offer are over-the-top.

Like the previously mentioned commercial DSP, the *MiniDSP 2x4 HD* uses a more powerful *ADSP21489* SHARC audio processor [71]. This high-definition (HD) version also has USB audio streaming thanks to the *XMOS Xcore200* [72] and also a Toshiba Link (TOSLINK) optical digital input. Both MiniDSP products are pictured in the figures on the facing page.

3.3. Integrated Development Environments from Analog Devices

ADI offers very competitive products and has a great framework to build on. Here we present the three IDEs provided by ADI.

3.3.1. VisualDSP++

“VisualDSP++ for Blackfin, SHARC, and TigerSHARC processors is an easy-to-install and easy-to-use integrated software development and debugging environment that enables efficient management of projects from start to finish from within a single interface.” [73]

VisualDSP++ has a C/C++ compiler, assembler, linker and loader that can emulate programs on real hardware or simulate them on the computer with controlled data. The software is not free but can be evaluated for a period of 90 days. The licensing cost for the SHARC development software is around 4200 \$ and up to 18 000 \$ for the development and debugging license.

3.3.2. CrossCore Embedded Studio [75]

“CCES is a world-class integrated development system (IDE) for the ADI Blackfin, SHARC and ARM processor families.” [75] This software is quite similar to VisualDSP++ but it is more affordable, with an individual licensing cost down to 240 \$. It also offers a 90-day trial.

3.3.3. SigmaStudio

SigmaStudio is ADI take on a “what you see is what you get” (WYSIWYG) graphical IDEs. It supports all SigmaDSP audio processors and even some SHARC SoCs when installing an extension.

The main benefit of this software is that it allows engineers with little or no DSP coding experience to add quality digital signal processing to their designs. It offers a wide variety of signal processing algorithms integrated into an intuitive GUI, allowing the creation of complicated audio signal flows. The tool can help users lower their costs by reducing development time without sacrificing quality or performance [76].

We believe SigmaStudio is a successor of VisualAudio Designer, a no longer supported IDE from the early 2000s [77]. This software was able to translate a graphical audio dataflow into C source files VisualDSP++ could interpret.



Figure 3.5.: ADAU1701 Evaluation Board [68]

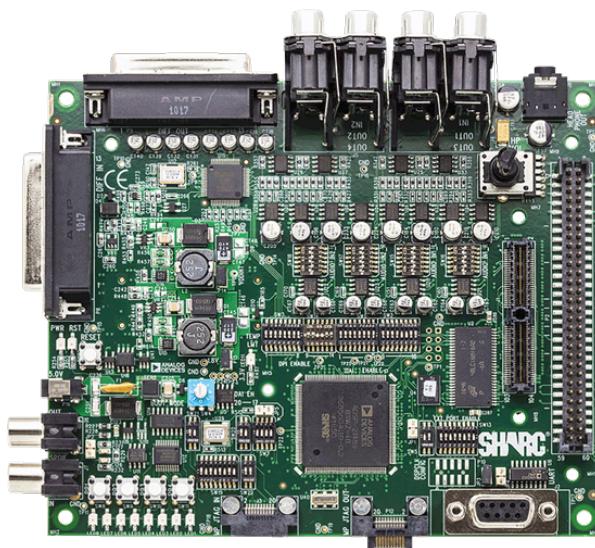


Figure 3.6.: The EVAL-21489-EZLITE evaluation board



(a) *miniDSP 2x4 kit*



(b) *miniDSP 2x4 HD kit*

Figure 3.7.: The two most sold DSPs by *MiniDSP* in DIY kit format

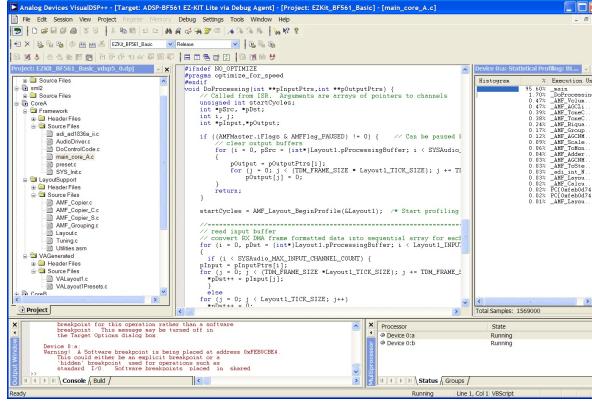


Figure 3.8.: The VisualDSP++ IDE [74]

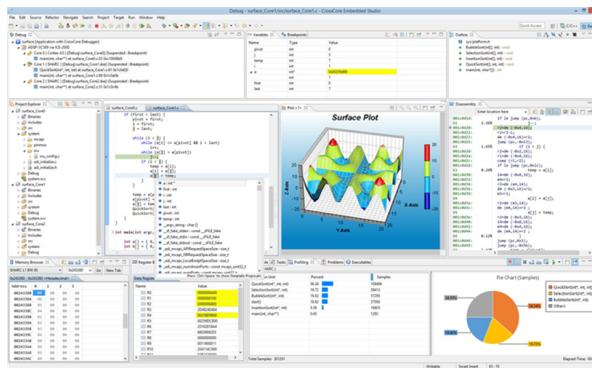


Figure 3.9.: CrossCore Embedded Studio screenshot



Figure 3.10.: A SigmaStudio project with what appears to be a 3-way crossover

3.3.4. Which is the best for students?

In the section 3.1.1 on page 7 we were searching for a graphical IDEs. From the information we have gathered, it is clear that neither VisualDSP++ or CCES are adequate solution for students. We will expand on this later. But before we make any blind choices, let us expand on how SigmaStudio.

SigmaDSP audio processors are not the only devices compatible with SigmaStudio. Even some SHARC processors are compatible. On the other hand, the Analog Devices Wiki has the “List of ICs Supported by SigmaStudio” — in which SHARC processors are not part listed — where we can see that different families of ICs are present [78].

SigmaStudio can interface directly with multiple evaluation boards and support a couple of USB to serial converters. The EVAL-ADUSB2EBZ, in short *USB*_i**, is a host In-Circuit Programmer (ICP) that interfaces with all SigmaDSP evaluation boards [79]. SHARC evaluation boards require an adaptor because they use a slightly different interface [80].

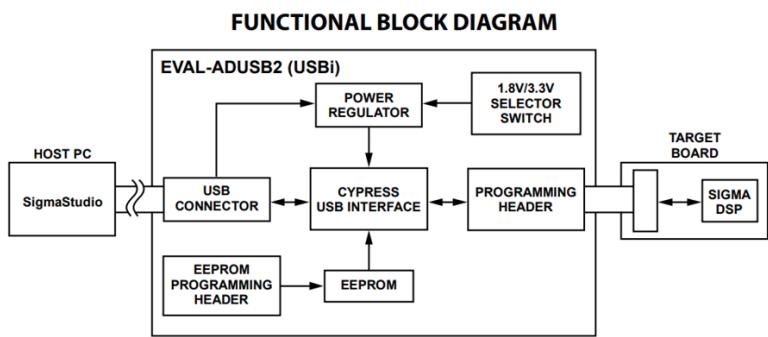


Figure 3.11.: Functional block diagram of the EVAL-ADUSB2EBZ USB to serial peripheral interface (SPI)/inter-integrated circuit (I^2C or I₂C) converter [81].

SigmaDSP audio processors do not have a SoC like SHARC processors do. Once we understand this distinction, we can see the inherent complexity difference that SHARC processors add. One could argue that FPGAs are capable of computing what DSPs can achieve. The previous statement is true, but in the context of this thesis and the curriculum of the ICT Systems engineering degree, FPGA design and programming are already covered in the subjects of *Computer Architecture* and *Embedded Systems* respectively.

In the scope of the subject of PDS, SigmaStudio provides the perfect solution for the students.

3.4. Choosing a DSP

Like we stated previously, SigmaStudio is capable of working with many devices but is specifically targeted to work with SigmaDSP audio processors. SHARC processors are not entirely compatible with SigmaStudio, and according to the installation manual, it is required to have installed CCES along the *SigmaStudio for SHARC* extension [82]. CCES's license can be as a 90-day trial or through Analog Devices' Engineering University Program [63].

Regarding the families of processors on page 13, *Blackfin* embedded processors will be discarded for the sole reason that they are not compatible with SigmaStudio, making them unsuitable for our end purpose.

SHARC processors would give the best of both worlds with a high expandability and SigmaStudio compatibility. The problem is the high complexity of development of a PCB on this SoC. The application notes [83] and the 1300-page hardware reference manual [84] of the *ADSP-21489* prove the point. Thus, designing a didactic platform with SHARC audio processors is also out of the scope of this project due to the time constraints and the workforce of a single person, *i.e.* the author's.

Finally, although *SigmaDSPs* are not SoCs or MCUs, in that it cannot be programmed at a low level, the ease of programming through SigmaStudio makes it a perfect candidate for the users to quickly develop prototypes. This kind of workflow is valued by manufacturers because developers have a clear, visual representation to reference without second-guessing themselves [85].

3.4.1. The SigmaDSP audio processor line

As of the making of this thesis, there are 18 parts in the SigmaDSP audio processor line. To reduce the list we have made the following filtering criteria. We have begun by discarding the parts from the *ADAU* family because they are suited for advanced television products because they have an excessive amounts of inputs and outputs for developing AV switch.

Like with any other electronics components manufacturer there exists multiple part numbers with similar capabilities. We have discard all part numbers with lower capabilities, *e.g.* the *ADAU1702* which is lower-spec variant of the *ADAU1701* that has $1/2$ the program random access memory (RAM) and $1/4$ of the data RAM, but also $1/2$ the IPS [86].

Some audio processors do not have built-in CODECs, thus requiring external CODECs for processing analog signals. For this reason, any DSP that does not have have multiple SRCs internally would limit our processing dataflow to a single sample rate. Since ADI has many SigmaDSPs with multiple SRCs, we have decided to exclude the ones that do not have any SRCs.

On a final note, all components that have a BGA footprint have been avoided. This explained better in the appendix ?? on page ??.

Table 3.4: SigmaDSP audio processors [87]

Part Number	Memory (KWords)		Serial Ports		Stereo	Core		Delay	Self-boot?	Primary Function
	Data	Prog.	In	Out	SRCs	(Hz)	IPS	Pool		
ADAU1467	80	24	12	12	8	300M	6144 ms	1600	Yes	SigmaDSP
ADAU1462	48	16	12	12	8	300M	6144 ms	800	Yes	SigmaDSP
ADAU1442	12	4	9	9	8	172M	3584 ms	170	Yes	SigmaDSP
ADAU1401A							1024 ms	40	Yes	CODEC [†]
ADAU1701							1024 ms	40	Yes	CODEC [†]
ADAU1761							1024 ms	86	No	CODEC [†]

[†] 2 ADCs & 4 DACs

The audio processors that are capable of self-boot operation can be operated through an external electrically erasable programmable ROM (EEPROM). The CODECs are capable of operating in stand-alone mode without the need of additional peripherals to work, like DACs or ADCs.

3.4.2. SigmaDSPs with integrated CODECs

ADAU1401A & ADAU1701

The ADAU1401A² [88] and the ADAU1701 [89] are word-for-word identical in specifications, with the key difference of ADAU1401A having been approved for automotive applications.

We have previously seen this CODEC on page 13 in the *MiniDSP 2x4* product. With a 1024 instructions per cycle, the DSP core is advertised to operate at 48 kHz or 44,1 kHz, but doubling or quadrupling the core sample rate to up to 96 kHz and 192 kHz will reduce the instructions per cycle to 512 and 256, respectively. The same applies to the ADAU1401A.

ADAU1761 [90]

The ADAU1761 is similar to the ADAU1701, except for the removal of 2 DACs and having a limited sample rate of 96 kHz. It has a headphone jack detection circuit and a lower power consumption of 14 mW, compared to the 286,5 mW of the ADAU1701. This CODEC exists for integrating it in low power devices. The drawback is that it has a limited amount of GPIO pins, which makes expansion almost impossible because they are shared with the single serial data I/O port.

3.4.3. SigmaDSPs without integrated CODECs

ADAU1442, ADAU1445, ADAU1446 [91]

These processors feature a flexible audio routing matrix with 24-channel digital inputs and outputs and up to 8 stereo asynchronous SRCs (ASRCs) with a sample rate of 48 kHz and up to 192 kHz. It has a total of 9 serial data input ports and 9 serial data output ports, and a S/PDIF receiver and transmitter. It also features 12 GPIO pins, 4 of which are assigned to 10-bit auxiliary ADCs that can be used to read potentiometers and such.

The high flexibility of this line of processors could be useful for example to use it as an auxiliary accelerator processor in FPGA applications.

ADAU1462/3 & ADAU1467/6

The part numbers ADAU1462 & ADAU1467 mentioned in the title are the higher-spec variant of their counterpart. This is the case because the four products offer a mix and match of serial inputs and outputs ports. The ADAU1467 boasts a large memory and is the SigmaDSP audio processor with the largest digital audio delay pool. Delay pools are useful for example for acoustically synchronizing audio signals from the listeners perspective. The functional block diagram is shown on the next page.

²The ADAU1401A is a revised version of the obsolete ADAU1401.

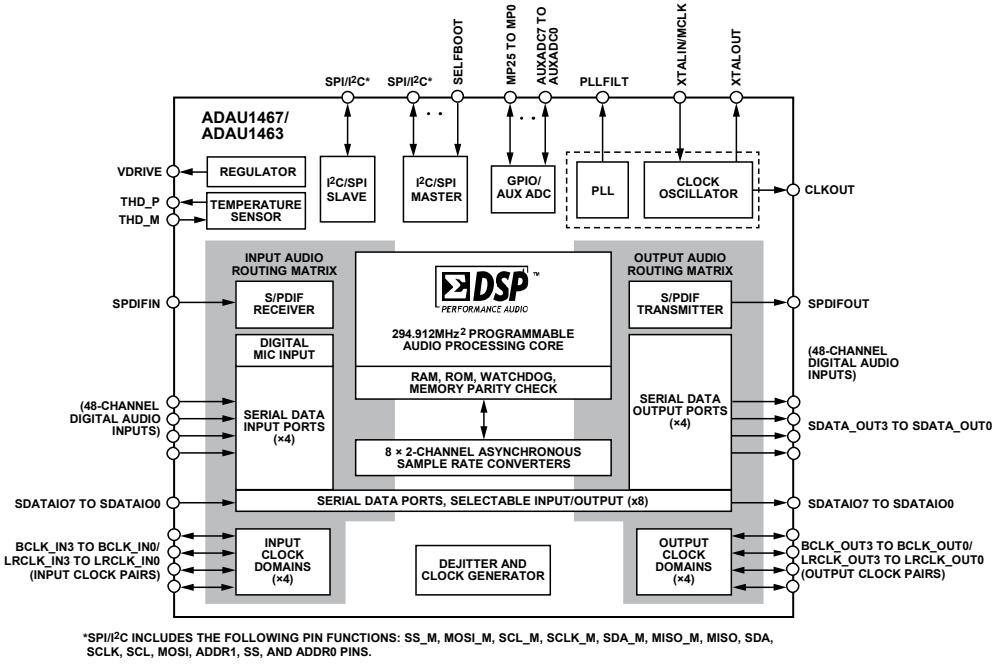


Figure 3.12.: Functional block diagram of ADAU1467

3.4.4. Choosing a DSP: Conclusion

To compensate for the lack of not having a low-level programmable core like in a SHARC SoCs, a FPGA or a MCU, we will not be choosing the SigmaDSPs with built-in CODECs because they have limited expandability.

The final pick will therefore be the **ADAU1467**, because out of the other DSP processors, this is the best in terms of specifications.

3.5. The ADAU1467 DSP

Like we explained, the ADAU1467 is a part that does not have CODECs, therefore inputs and outputs are consists of various ports where peripherals can be connected, *e.g.* ADCs, DACs, pulse-density modulation (PDM) microphones, or external S/PDIF transceivers. A single S/PDIF receiver and transmitter is built-in the chips routing matrix, since it just requires a single-ended digital signal. Nevertheless, with a grand total of 8 serial ports that can handle a total of *48 audio channels* simultaneously at 48 kHz, an uncountable number of projects should be possible.

We can see in the figure on the facing page that the inputs are routed directly to the DSP core, the ASRCs, and serial output ports. The internal audio routing of the ADAU1467 is too extensive to cover in this document alone, thus we will be focusing on the routing to the AD1937 CODEC via the outermost shell of the processor, through the serial ports.

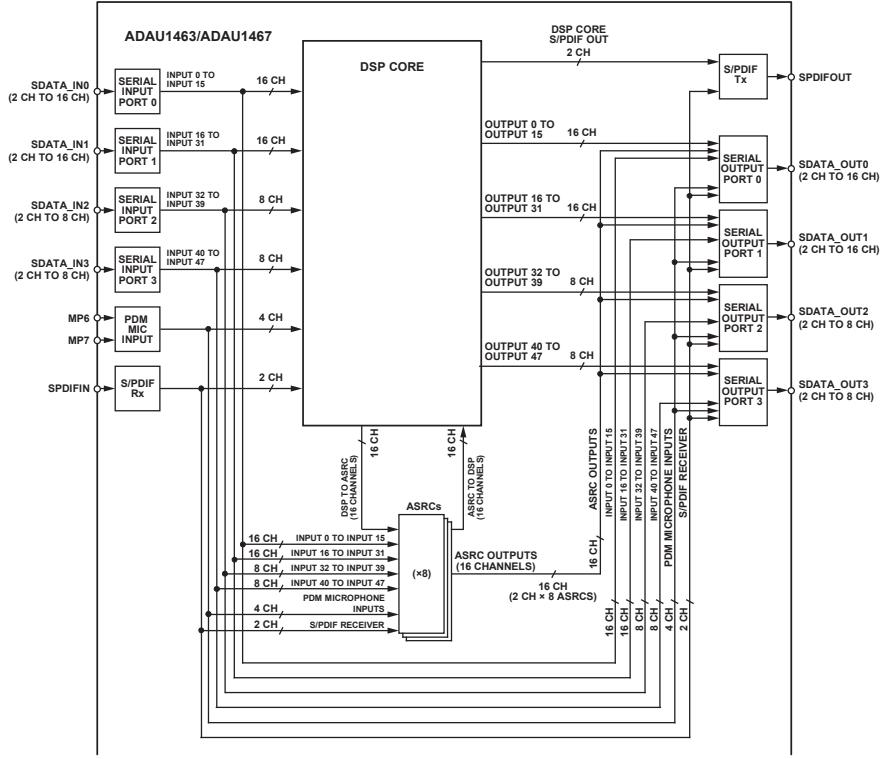


Figure 3.13.: External to internal audio routing overview of the ADAU1467

The ADAU1467 serial ports are divided in inputs and outputs: 4 input ports (**SDATA_IN**) and 4 output ports (**SDATA_OUT**). Additionally, 8 supplementary serial data pins (**SDATAIO**) can add extra data lines to the 8 serial ports. The reason behind this is, even if we have modes like TDM16 that can support 16 channels, most audio peripherals cannot use TDM and are limited to the two-channel I²S [92]. The extra supplementary data pins help to reduce number of serial ports that would be required to connect I²S-only devices.

The SDATA ports are grouped together in SigmaStudio according to the table on page 23. Each port group has at maximum 16 channels, configurable in the streaming modes of: I²S (2-channel), TDM4, TDM8, or TDM16. The latter is not available for SDATA ports 2 & 3 because these ports support a interface called “Flexible TDM”³.

The channel mapping for the inputs and outputs can be seen in table 35 and 36 of the ADAU1467’s datasheet [93].

3.5.1. The I²S serial bus

The I²S serial bus connects ICs together that need to send digital audio between them. Although the name of this protocol suggests it is related to inter-integrated circuit (I²C or I2C) [94], it does not offer bidirectional multi controller/target serial communication. Both protocols have a similar name because they are made by Philips Semiconductors, in the eighties.

³An optional mode that provides byte addressable data placement in the input and output data streams on the corresponding serial data input/output pins.

The protocol defines a straight-forward way to sent two N -bit audio samples using TDM using just 3 single-ended lines:

1. A continuous serial clock (SCK) line, typically written as bit clock (BCLK), which triggers a new bit on each falling edge.
2. A word select (WS) line, but better represented by left-right clock (LRCLK), which toggles every N bits what channel is being written to (master mode) or read from (slave mode).
3. A SDATA line, which transmits the audio samples in big endian format, *i.e.* most significant bit (MSB) first.

Optionally, to synchronize the master device with the slave devices, a third clock signal called master clock (MCLK) is added. This clock line operates at $256 \times LRCLK$ (12.23 MHz for a sample rate of 48 kHz). All the clocks operate with a continuous square wave with 50% duty cycle.

When LRCLK triggers on a falling edge, the left channel is being accessed, and a rising edge means the right channel is being accessed. This is equivalent to having divided the time required to send both channel samples in two slots, which would be equivalent to TDM2. Other variants of I²S use many more TDM slots to send the from 4 and up to 16 separate audio channels. A timing diagram for a stereo I²S bus is provided in the figure below.

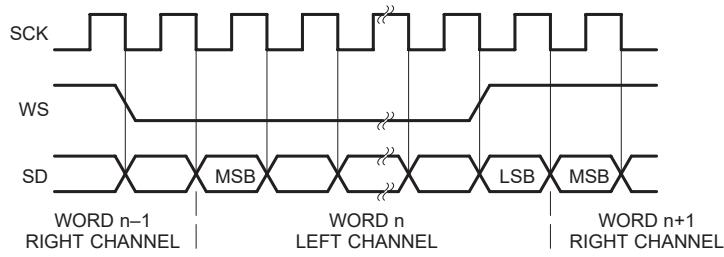


Figure 3.14.: I²S timing diagram

3.6. From Digital to Analog, and Vice Versa

Because most of the DSPs presented did not have built-in CODECs, many applications would require processing of analog signals. Since we are developing a didactic platform, we believe *the minimum that should be provided* is a set of analog inputs and outputs. This is the reason we will present a commonly used family of CODECs used with CODEC-less DACs.⁴

3.6.1. The AD193x Family of CODECs

The AD1937, AD1938, and AD1939 [95] from Analog Devices have 4 ADCs and 8 DACs each. All three have differential ADCs. The AD1937 and AD1939 DACs are also differential, except for the AD1938, which is single-ended. We can see the functional block diagram for the AD1939 in the figure on the next page.

⁴The ADAU1467 has 8 auxiliary ADCs of 10 bits of accuracy. They are intended to be used with control signal inputs, such as potentiometer outputs or battery monitor signals, and not audio signals of high sample rate.

Table 3.5: Unidirectional serial port channel mapping in SigmaStudio

Serial Port	Channels
SDATA_IN0 or SDATA_OUT0	0 to 15
SDATA_IN1 or SDATA_OUT1	16 to 31
SDATA_IN2 or SDATA_OUT2	32 to 39
SDATA_IN3 or SDATA_OUT3	40 to 47

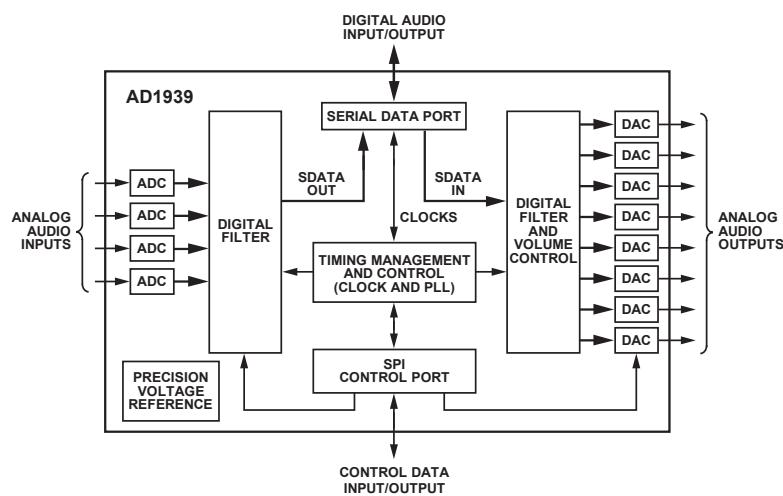


Figure 3.15.: Functional block diagram of the AD1939 CODEC

3.6.2. The AD1939⁵

This CODEC is gets connected to a MCU or a DSP via multiple serial data ports. The digital data from each ADC and DAC contains a single audio channel (or slot, in TDM terms). Since this CODEC can communicate in I²S, the channels are grouped in pairs and can be read from up to 6 serial data pins.

The serial ports of the audio path are capable of running in three different modes: stereo (I²S), TDM, or auxiliary (AUX). By default, the mode is set to I²S which is the standard serial bus for most applications. The mode can be changed through the communication with the device via the SPI/I²C control port. This CODEC can also operate in standalone mode, where no external communication is required and the device operates in slave mode. In this configuration, the AD193X operates at a sample rate of 48 kHz in stereo mode.

The maximum sample rate of the AD193X is 192 kHz with 32 bit words.

Stereo mode

In stereo (I²S) mode, each pair of channels gets its own SDATA pin. The four DACs input pins (**DSDATA**) share the same serial bit clock (**DBCLK**) and a left-right framing clock (**DLRCLK**), and the two ADC output pins (**ASDATA**) share the same serial bit clock (**ABCLK**) and left-right framing clock (**ALRCLK**).

The drawback is, that aside from the two serial data ports that must be used at all times because the I²S protocol sends data unidirectionally, four extra pins are required, which adds to the total: 4 **DSDATA** pins and 2 **ASDATA** pins. The benefit is that in this mode, the full sample rate of 192 kHz at 24 bit is available.

AUX mode

This mode repurposes some of the SDATA pins so we can add to the system up to two additional stereo DACs and two stereo ADCs, increasing the channel count up to 8 inputs (ADC) and 12 outputs (DAC).

TDM mode

Lastly, in TDM mode, like the name implies, all the input channels are sent through a single SDATA pin by the means of time division multiplexion. This limits the sample rate to 48 kHz, 44,1 kHz, or 32 kHz, because the 16 × 32 bit words are sent each LRCLK. This requires 512 BCLKs, or 24,58 MHz, which coincides with the maximum MCLK of the CODEC.

The benefit of running in this mode is that some pins previously used for stereo mode signals are not unused to talk with the DSP, but still have a purpose in the system. Because in TDM16 mode the total number of slots is 16 and the CODEC by itself does not have neither 16 inputs or outputs, the leftover pins are used to chain a second AD193X to the system.⁶ Now, with an auxiliary CODEC, the maximum number of channels is of 24: 8ADCs, and 16DACs.

⁵All the text from this section also applies to the AD1937 and AD1938.

⁶It does not have to be a codec of the same family, as long as it can understand the same serial port protocol.

The other benefit of this mode is that there are still two pins left in the system that are used to increase the throughput of the DAC channels. In this “dual-line mode”, the DAC SDATA pins can run in TDM4, allowing a sample rate of 192 kHz (maximum 8 channels because $192\text{kHz} \cdot 32b \cdot 4 = 24.576\text{MHz}$), or in TDM8 allowing a sample rate of 96 kHz (maximum 16 channels, because $96\text{kHz} \cdot 32b \cdot 8 = 24.576\text{MHz}$).

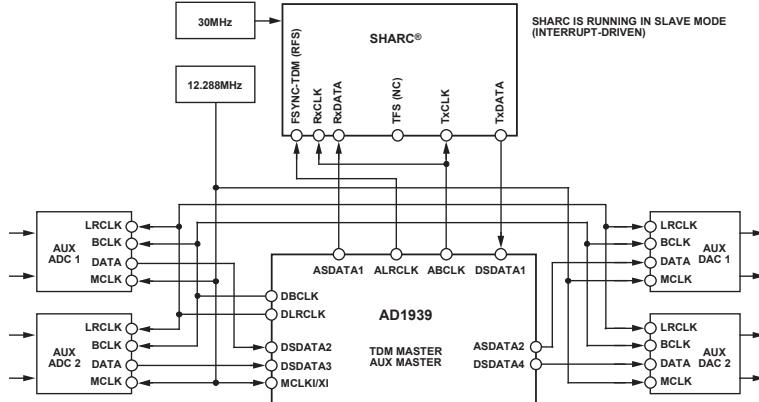


Figure 3.16.: Example of AUX Mode Connection to SHARC (AD1939 as TDM Master/AUX Master Shown)

Moreover, the AD1939 is able to handle all these modes in master or slave mode, *i.e.* the clock sync signals are received from a master device to itself, or sent by itself to a slave device. The figure above shows one such configuration, where the inputs and outputs clocks of the SHARC processor are received from the AD1939.

For a thorough explanation on all the available modes and timing diagrams, visit section “SERIAL DATA PORTS – DATA FORMAT” on page 15 from the AD1939’s datasheet. [96]

Difference between the three CODECs

The difference between the AD1937, AD1938, and AD1939, aside from the differential or single-ended outputs, is that the AD1937 is the only CODEC with an I²C control port. The fact that the control port has a different protocol means a lot to the design of a DSP platform like ours.

The ADAU1467 features in total 3 control ports. The first control port is out of our interest because it is the slave I²C/SPI combined port, which is used to program the device. On the other hand, the remaining two are the master control ports, which are a combined I²C/SPI port, and a secondary I²C port.

The master control ports can be used to boot the ADAU1467 from an external serial EEPROM, boot and control external slave devices, or to read and write to an external SPI RAM or flash memory. We will include an external *serial* EEPROM because the ADAU1467 program memory is volatile. Therefore, we are limited to control external devices through the remaining I²S port found in the multipurpose pins SCL2_M/MP₂₄ & SDA2_M/MP₂₅.

This makes us to have to choose the AD1937, which is the only CODEC with an I²S control port.

Mild disclaimer for the reader: the only reason we have gone and used Analog Devices CODECs is because they are compatible with SigmaStudio. To be able to start the board in selfboot mode, requires the developer to include the compiled code of the external slave devices in a master control port block inside through SigmaStudio. We are inclined to believe that manually generated code should be able to be generated and loaded for an unsupported device [97, 98].

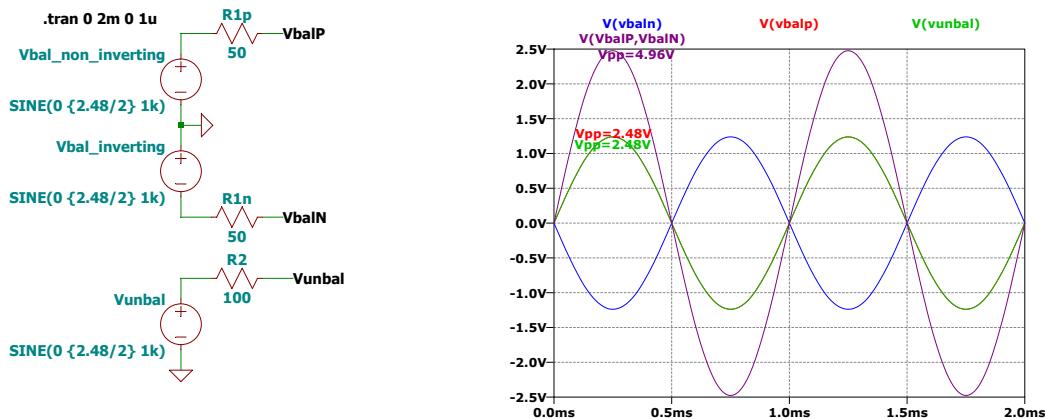
3.6.3. Single-ended and differential signals

In context, single-ended signaling consists of driving a single conductor with signal which is referenced to a fixed voltage. Differential signaling consists of the transmission of a single-ended signal as a symmetrical pair, *i.e.* the signals are equal in magnitude but of opposite polarity.

“The symmetrical signals of differential signaling are often referred to as balanced, but this term should be reserved for balanced circuits and lines which reject common-mode interference when fed into a differential receiver. Differential signaling does not make a line balanced, nor does noise rejection in balanced circuits require differential signaling.” [99]

Differential signals are preferred in data transmission because of the resilience to noise, but most consumer grade audio products use single-ended signals. Either way, we will check the differences between the AD1938 and AD1939 to better understand these concepts and to present some examples.

The single-ended outputs of the AD1938 have a full-scale output voltage of 0.88 VRMS or 2.48 Vpp, and the differential outputs of the AD1937/9 have a full-scale output voltage of 1.76 VRMS or 4.96 Vpp, exactly double of the AD1938. The equivalent schematic and diagram are show in the figures below



(a) Schematic of AD1937/9 (top) and AD1938 (bottom)

(b) Simulation at 1 kHz for 2 ms

Figure 3.17.: Thevenin’s equivalent of the AD193X DAC outputs at 0 dBFS

To achieve a *line out* level, *i.e.* the fixed-voltage level used between audio devices specified in the table on the facing page, we can see the AD193X family can achieve at least the consumer audio line level output.

Table 3.6: Line levels and their approximate nominal levels [100, 101]

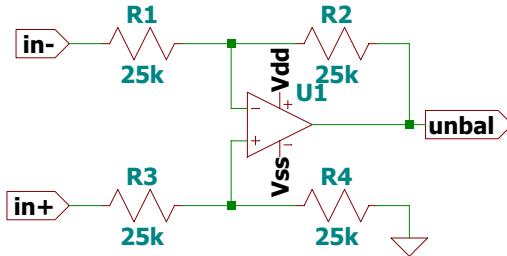
Audio use	Nominal Level	VRMS	Vpk	Vpp
Professional	4 dBu	1,23	1,74	3,47
Consumer	-10 dBV	0,32	0,45	0,89

3.6.4. Conversion from balanced to unbalanced⁷

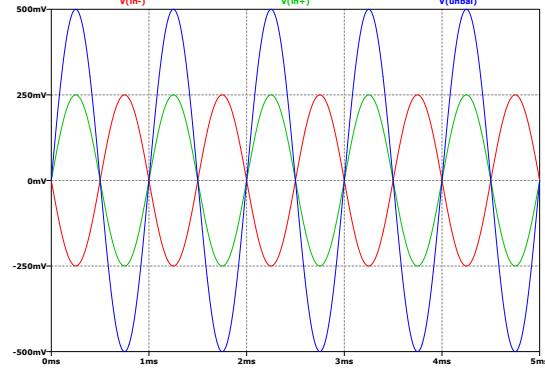
Consumer audio devices usually have single-ended I/O. Conversion from differential to single-ended can be achieved in different ways: [102, 103, 104]

1. Using a first order high-pass RC filter to decouple the headphone amplifier (HPA).
2. Using operational amplifiers (op-amps). See figure below
3. Using a transformer.

The instrumentation amplifier is a common configuration in which a differential signal is converted to a single-ended signal. This configuration needs tight tolerances from the resistor pairs R1/R2 and R3/R4 to reduce imbalances in the output.



(a) Schematic



(b) Plot of differential inputs and single-ended output

Figure 3.18.: Instrumentation amplifier used as a line driver

Based on the research we have conducted on the “Op Amp Applications Handbook” [105], we have concluded that using an op-amps to drive low impedance loads is not an optimal idea. Due to the nature of consumer audio devices, *e.g.* earphones and headphones, many have impedances as low as 16Ω which are hard to drive with op-amps. On the other side of the spectrum, high impedance loads are easy to drive since they do not require much current.

$$I = \sqrt{\frac{P}{R}} \quad \begin{cases} 25mA, & \sqrt{10mW/16\Omega} \text{ (earphone);} \\ 4mA, & \sqrt{10mW/600\Omega} \text{ (headphone).} \end{cases} \quad (3.1)$$

$$V = \sqrt{P \cdot R} \quad \begin{cases} 400mV, & \sqrt{10mW \cdot 16\Omega} \text{ (earphone);} \\ 2449mV, & \sqrt{10mW \cdot 600\Omega} \text{ (headphone).} \end{cases} \quad (3.2)$$

⁷We will use interchangeably *unbalanced* with *single-ended*, and *balanced* with *differential* throughout the document.

The equations on the preceding page compare the current and voltage required to drive very low impedance earphones with very high impedance headphones. The 600Ω impedance was chosen because it is the reference impedance used typically in audio related studies. We can see how each device has completely different needs.

Most op-amps are unable to drive low impedance loads because they have a limited output current in the order of tens of milliamps. For example, the TL072 op-amp has a short-circuit current of $\pm 26\text{mA}$. To reduce the load on a single op-amp, splitting the load is a viable solution that can be achieved by paralleling two or more op-amps. The figure below show one way of implementing it.

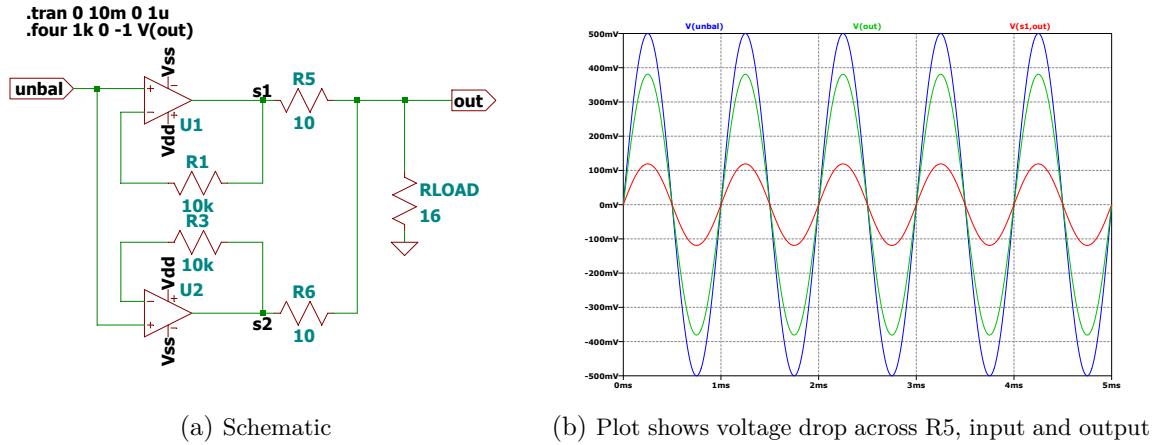


Figure 3.19.: Two voltage followers in parallel

The two voltage followers act independently to split the load. The 10Ω resistors protect the op-amps from being shorted. These resistors also prevent the op-amps from sinking current caused by manufacturing tolerances. Using resistor arrays and dual/quad op-amp chips can somewhat solve this problem. The voltage drop across the resistors can be eliminated through a closed feedback loop. See figure above for a simulation.

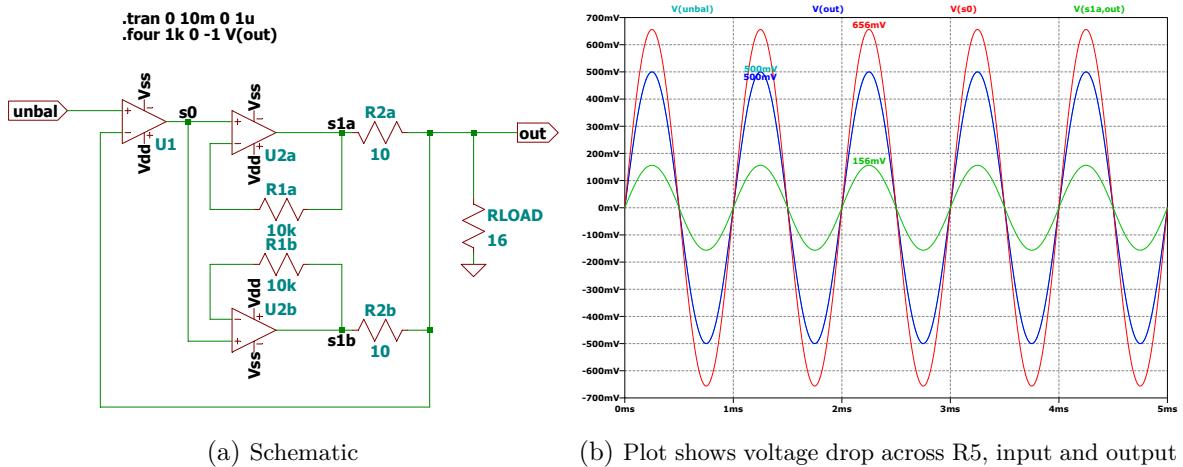


Figure 3.20.: Two voltage followers in parallel with an error amplifier

The drawback of using many op-amps in parallel is the overall cost of the components. We also have to remember that this design requires a symmetrical power supply. The circuits mentioned previously can be operated from a single supply using a virtual ground [106], although a capacitor is required to block DC coming from the virtual ground of a single-supply amplifier.

A better solution for driving low impedance loads

A better solution than using op-amps in parallel or using discrete components like in a diamond buffer [107], would be to use an integrated headphone amplifier, like the MAX9722A [108].

The MAX9722A uses an architecture to eliminate the DC-blocking capacitor required on the output of traditional headphone amplifiers [109]. In the previous circuit, a DC blocking capacitor would have been required to protect the input and output from the bias voltage. This capacitor would need to have a large value to get a flat frequency response down to 20 Hz.

The MAX9722A also integrates an undervoltage-lockout (UVLO) and shutdown control circuit to power down the IC when the device has an unexpected power drop or needs to be shut down to save power. It also has a built-in “click-and-pop” suppression circuit, a common problem with audio equipment when connecting a device to a powered source. These features eliminate the need of external circuitry that would otherwise be hard to implement.

The benefit of using the MAX9722A over the op-amp solution is the lower cost and design complexity. Moreover, the MAX9722A can be operated directly from a CODECs differential output, like that of the AD1937/9. Moreover, the MAX9722A is also compatible with the MAX97220A with some minor schematic differences [110], increasing the availability when having to order either chip and the stock is scarce.

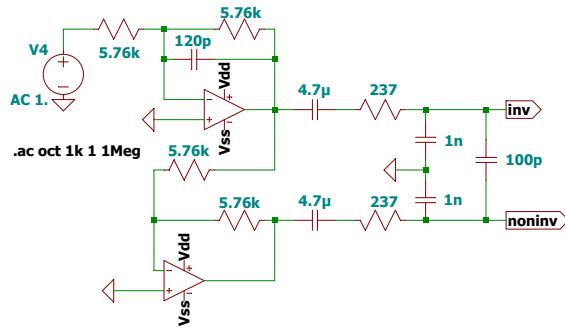
3.6.5. Conversion from unbalanced to balanced

Like we explained, the AD1937/9 uses balanced ADCs inputs, but the user would probably use unbalanced signals. The AD1937's datasheet provides a recommended ADC input filter that converts from single-ended to differential. The circuit has been simulated in *LTS spice XVII* and the bode plot is provided in the figure on the next page.

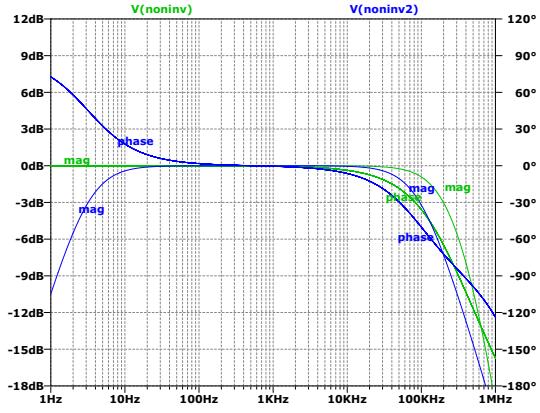
We have simulated both designs to compare the slight difference the component values and the effects of the DC blocking input capacitor do to the phase-magnitude plot. The ADC filter uses two inverting unity gain buffers stages, the first having a low-pass filter. On the recommended design from the datasheet, the low-pass filter cut-off frequency is about 200 kHz, and the evaluation board cut-off frequency is about 100 kHz. Aside from that, the evaluation boards DC blocking capacitor has a cut-off frequency around 3 Hz.

3.6.6. Bypassing the DSP

One feature that we would like have, based on the EVAL-ADAU1787Z [112], is a switch that bypasses the CODEC and DSP completely and redirect the input to the output. The EVAL-ADAU1787Z is able to have this feature because the ADAU1787 only has two analog outputs, the internal DACs, which are wired externally to the bypass switch to one of the stereo TRS jacks. If we wanted to implement the same in the AD193X, because we have 8 DACs and 4 ADCs, at most 2 stereo channels could be bypassed.



(a) Schematic of the AD1937 datasheet ADC driver



(b) Bode plot with component values from the datasheet (noninv) and EVAL-ADAU1467Z [111] (noninv2) documents

Figure 3.21.: ADC driver recommended filter by the AD1937's datasheet ()

The switch used in the EVAL-ADAU1787Z is a 4PDT sliding switch, *i.e.* a switch that actuates mechanically four switches (4 poles, or 4P) with two positions (dual throw, 2T or DT) at the same time. Two poles are used for the stereo audio channel routing while a 3rd and 4th poles are paralleled together to the active-low reset generator. The reason behind this choice is because the DSPs can be powered down while in the bypass switch is activated.

To save space, instead of using a 4PDT switch we can use a latching SPDT switch to generate a logical signal that would power a audio switch, also called a bilateral switch. These are electronic component that behaves in a similar way to a relay, but has no moving parts [113]. An analog switch is composed internally of a transmission gate with an on-resistance usually of less than $100\ \Omega$, some even reaching the $m\Omega$ territory. For instance, the ADG884 is a dual 2 : 1 MUX/SPDT audio switch with an on-resistance of $0,33\ \Omega$ [114].

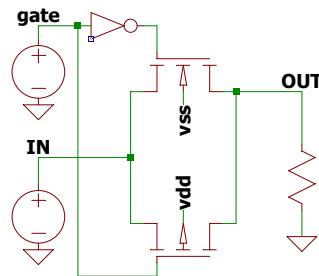


Figure 3.22.: A transmission gate with a symmetrical power supply

We will not be using a bilateral switch because there exist subminiature 4PDT pushbutton switches that require less space than sliding switches, thus eliminating the circuit proposed [115].

The TOSLINK could also be bypassed by wiring the digital output of the receiver to the digital input of the transmitter [116]. But for this we would need a 1 : 2 demultiplexer and a 2 : 1 multiplexer to passthrough from one TOSLINK jack to the other, like in the figure below.

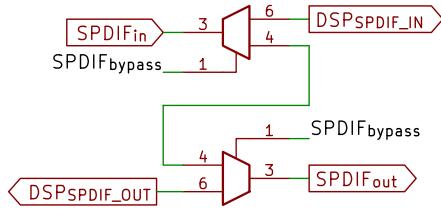


Figure 3.23.: S/PDIF digital TOSLINK bypass using a multiplexer and a demultiplexer

But because the digital S/PDIF will probably not be used a lot, we will exclude this circuit and recommend using a digital bypass through SigmaStudio instead.

4. Development

Now that we have got a general guideline, we will begin with the development of the schematic for the development board, while explaining our selective process for the design.

4.1. Our KiCAD Project

Since we will be using KiCAD [24] to design the schematic and PCB layout, we should inform about some of our decisions that will improve the readability of the project files.

4.1.1. KiCad library convention

Before we begin with the PCB design, we would like to introduce the KLC. [citekicad:klc](#) This website gives a set of *guidelines*, not rules, to be able to contribute to the official KiCad libraries. These guidelines define things such as: symbol naming, pin grouping and position.

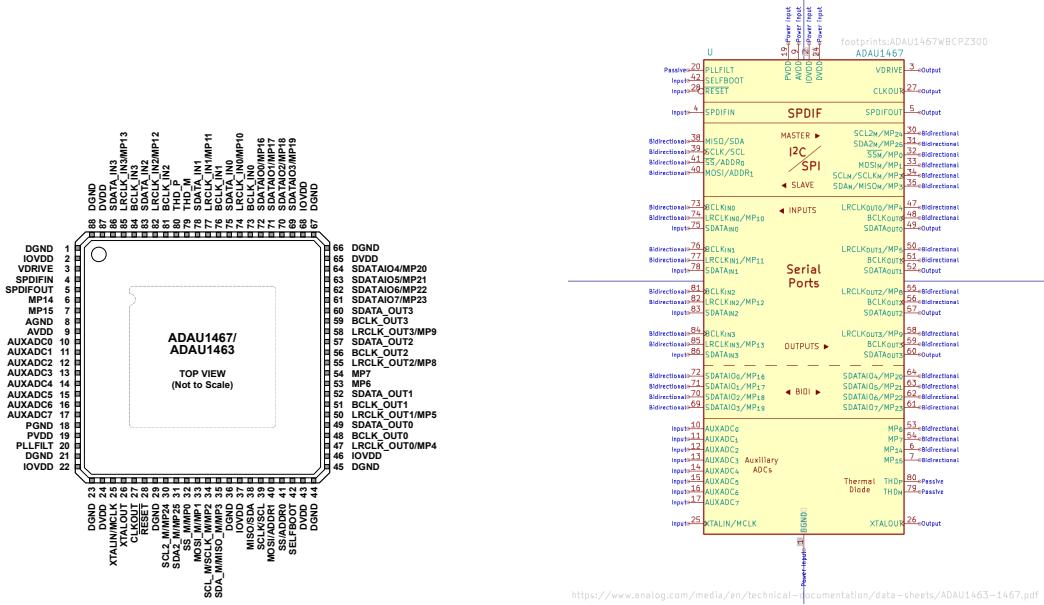
KiCAD includes a couple of ADI DSPs, like the ADAU1701. Using the KLC and the datasheet, we will create a the symbol for the ADAU1467. We will need to create a new symbol library, the we can copy one of the already KiCad library convention compliant symbols to begin as a template, or even better, we could use an online symbol and footprints library like UltraLibrarian [25].

We will be using these guidelines to improve the design of our schematics. The resulting symbol for the ADAU1467 is shown in the figure on the next page, and for the AD1937 in the figure on page 35.

4.1.2. The PCB guideline we have followed

As a general rule of thumb, we have made a step by step guidelines that should help with the design of our project. There are steps that are not covered on this list that will appear along the design of the development board. We will cover them whenever they come chapters or sections.

1. Schematic design:
 - 1.1. Create a schematic and define the page layout and information.
 - 1.2. Placing the core of the design.
 - 1.3. Design the audio interfaces.
 - 1.4. Design any remaining control interfaces
 - 1.5. Design power management circuit.
 - 1.6. Annotate schematic.
 - 1.7. Perform electrical rules check (ERC)
 - 1.8. Assign footprints while simultaneously choosing components.
 - 1.9. Iterate until satisfied...
2. Layout design:



(a) Topdown view form the datasheet

(b) Our KLC compliant symbol

Figure 4.1.: ADAU1467 symbol with helper labels

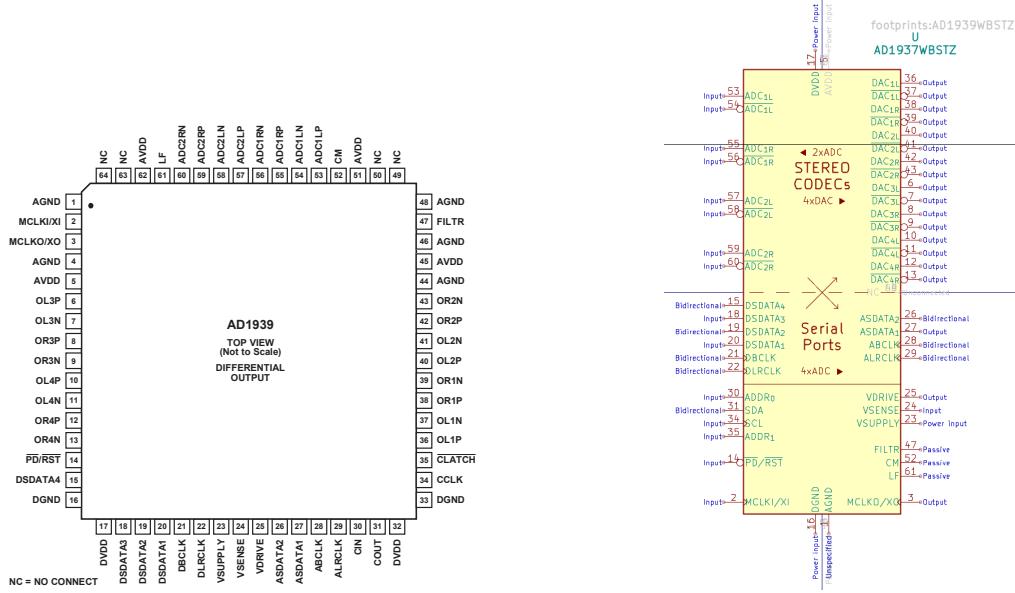
- 2.1. Modify the board setup, *i.e.* the layer stackup and manufacturing constraints.
- 2.2. Choose placement of outermost interfaces and connectors.
- 2.3. Component placement and routing.
- 2.4. Check nets and constraints.
- 2.5. Also move silkscreen text and add notes where needed.
- 2.6. Iterate until satisfied...
3. Make any changes to schematic if necessary.
4. Generate PCB fabrication files.
5. Add any missing 3D models and make a 3D render of the PCB.

4.2. Power management

The ADAU1467 requires a supply voltage of 3,3 V in the analog and digital power input pins. The SigmaDSP core also requires a digital voltage of 1,2 V which is regulated externally by the chip through a pass transistor, to lower the power dissipation of the device.

The AD1939 will also use the same supply voltage of 3,3 V, and the communication will happen at this level too. The CODEC also has an external voltage regulator, but since it is used to regulate an input voltage of 5 V down to 3,3 V, we will not use it.

Instead we will be using a low drop-out (LDO) regulator for the whole system. The LDO we have chosen is the NCV8186BMN330TAG. This very small form factor fixed voltage regulator, with a dropout voltage of just 100 mV at 1 A. It also features a built-in current protection circuit, through a feedback pin, plus it also has a chip enable pin. The *B* variant we have chosen does not have an output discharge gate, for when enable pin is pulled low and the output needs to be quickly discharged, because we will not be using it.



(a) Topdown view from the datasheet

(b) Our KLC compliant symbol

Figure 4.2.: AD1937 symbol with helper labels

Other than than, the current output should be sufficient with all the bulk capacitors, since the maximum current consumption of the ADAU1467 is around 800 mA, and the AD1937 pulls typically 429 mW at $256 \times f_s$ (48 kHz), or 130,03 mA.

The remaining high current consumption components, which are the operational amplifiers, will be driven by the power supply input, which will be of 5 V.

4.2.1. Bypass capacitors

The ADI recommends in the ADAU1467 datasheet to decouple each analog and digital power supply pin to its nearest appropriate ground. The datasheet also gives a recommended layout which we will use on the layout design, on page 205. More or less the same procedure will be used for the other components. All of the bypass capacitors will be placed on a separate sheet to prevent clutter, because bypass capacitors are usually recommended on every power input.

4.2.2. Manual reset and voltage supervision

The board also includes a voltage supervisor for the 5 V supply. The bulk capacitors should hold the charge for a few hundred milliseconds. When the power is removes, the 5 V supervisor will trigger a high-level signal that can be read through the multipurpose pin 7 (MP7). This trigger can be used to save to the EEPROM any necessary information before the power-down occurs.

The writeback function is not actually necessary, but can be used for demonstrative purposes in a laboratory work. By polling each second, for example, we can write to the EEPROM the required data without the need of the writeback circuit [117].

4.3. Interface with a computer

Like any microcontroller unit, the development board must have an interface to communicate with a computer.

One example of a USB-to-serial converter is the Arduino Uno R3. This educational board houses the ATmega328p MCU which is programmed through the universal synchronous and asynchronous receiver-transmitter (USART) port. The communication is handled by a secondary MCU, the ATmega16U2, which has a USB controller. This board has been flashed during manufacturing for the ATmega16U2 to work as a USB-to-serial controller. The user could misuse the development board and erase the firmware from the ATmega16U2 [118].

Specialized ICs, like the FT232RL by Future Technology Devices International (FTDI) [119], exist to handle the entire USB protocol on the chip and eliminate the need of programming specific firmware [120]. This device is extremely common for adding a USB serial port on many projects.

Most of Analog Devices DSPs interface through a I²C or a SPI port.¹ This port is referred to as *USBi* and it is based on the original Aardvark's host adapter [121]. Both the Aardvark and ADI EVAL-ADUSB2EBZ host adapter use a USB-to-I²C/SPI host controller. The latter uses a Cypress² CY7C68053 USB microcontroller [79].

Using the *USBi* interface is out of the question because it costs around 95€ and similar implementations with a Cypress MCUs are available. *Sure Electronics WONDOM ICP1* is an in-circuit programmer applicable for *WONDOM* products integrated with the ADAU1701 DSP [123]. This programmer uses the Cypress [123]. CY7C68013A [124]. The same MCU is sold under the function of a logic analyzer for as little as 5€ from Chinese wholesalers [125]. This board has been confirmed to work with SigmaStudio [126], although ADI will not provide support for the firmware of the in-circuit programmer. [127]

The problem with the CY7C68013A USB MCU is that the voltage levels are not compatible with some devices, therefore voltage level translators would be required. The *USBi* programmer solves this by using the CY7C68053 USB MCU, which has a 1,8 V core and 1,8 V to 3,3 V I/O.

We could use the CY7C68053 but there is a catch. This MCU has a very thin profile fine pitch BGA (VFBGA) footprint that has a pitch of 0,5 mm and 0,3 mm diameter balls. The problem with this footprint is that the spacing between balls is around 0,2 mm and most if not all PCB manufacturers who offer small quantity prototyping will quote large sums of money because the constraints are considered to be of high manufacturing difficult.

Using the information from the blogpost [126] and we have confirmed that the CY7C68013A breakout board works and it is detected by SigmaStudio. Using the schematic provided by [128], we have confirmed that after inserting status LED in the adequate positions, the LED blink when trying to program a dummy project on SigmaStudio, as seen in the figure on the next page.

We have provided a walkthrough on how we installed the firmware, in the appendix on page 67. To circumvent the high cost of the *USBi* programmer, we will develop an adapter that will go on top of the CY7C68013A "logic analyzer" breakout board.

¹According to SigmaDSP v4.7 Toolbox, some evaluation boards use USB and other even use the TCP/IP protocol stack.

²Cypress Semiconductor Corporation was acquired by Infineon [122]

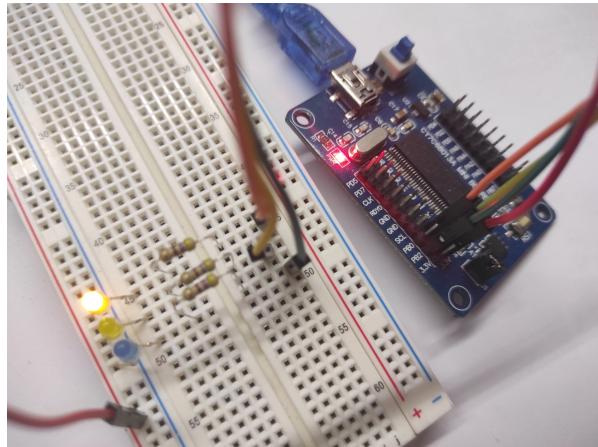


Figure 4.3.: The logic analyzer breakout board flashed with the USBi firmware, confirmed working by the status LEDs being lit

4.3.1. The adapter for CY7C68013A to Aardvark/USBi programmer

The pinout of the Aardvark and Analog Devices USBi in-circuit programmer is as follows in the on the next page.

The main difference, aside from the naming scheme, is that the USBi interface includes four additional chip select (CS) lines in total. This is useful to be able to program up to 5 SPI slave devices at the same time. The Aardvark header does not offer the pins 1, 2, 13 & 14. Both use an IDC-10 female header, sometimes called FC-10, and has pin pitch of 2,54 mm.

For the voltage level translator, we were going to use the TXB0108 components from the KiCAD library, but decided against it. The application note from Texas Instrument,[129] warns about the driving capabilities of the different families of voltage level translators, TI offers. The TXB family in particular, can have problems with I²C communication when devices communicating in the bus interface work against each-other by pulling the output low while the other devices are pulling the output high. This problem is solved by using voltage level translator with open-drain output, like the LSF010x or TXS010x [ti:txb, 130].

Please visit the appendix on page 67to view the full schematic and the 3D model incrusted in this document.

Table 4.1: Aardvark/USBi in-circuit programmer header interface

Pin	Aardvark	EVAL-ADUSB2EBZ (USBi)
1	N/A	$\overline{\text{CS2}}$
2	N/A	$\overline{\text{CS3}}$
(1) 3	SCL	SCL
(2) 4	GND	USB_CLK
(3) 5	SDA	SDA
(4) 6	+5 V	+5 V
(5) 7	MISO	COUT
(6) 8	+5 V	$\overline{\text{RESET}}$
(7) 9	SCLK	CCLK
(8) 10	MOSI	CDATA
(9) 11	SS	$\overline{\text{CS1}}$
(10) 12	GND	GND
13	N/A	$\overline{\text{CS4}}$
14	N/A	$\overline{\text{CS5}}$

4.4. Selecting the input clock

Both, the ADAU1467 and AD1937 require a clock source for their cores to work.

4.4.1. The ADAU1467 requirements

The ADAU1467 core works at frequencies between 152 and 294,91 MHz. The clock source is connected directly to the XTALIN/MCLK pin, or alternatively, the internal clock oscillator can drive an external crystal. The internal crystal oscillator driver is designed to work with crystals that resonate between 12 or 24 times less than the nominal system clock. If this is 147,46 MHz or 294,91 MHz, this frequency is 12,29 MHz.

“For systems that are intended to operate at a 48 kHz, 96 kHz, or 192 kHz audio sample rate, the typical master clock input frequencies are 3,07 MHz, 6,14 MHz, 12,29 MHz, and 24,58 MHz. The flexibility of the PLL allows a large range of other clock frequencies as well.”

We are not trying to use unusual values, so we will stick to the standard input clock frequencies. Although we do have to know that “the number of instructions that can be executed per sample is equal to the system clock frequency divided by the DSP core sample rate. However, the program RAM size is 8192 words; therefore, where the maximum instructions per sample exceeds 8192, subroutines and loops must be used to make use of all available instructions.” This is important, because operating at the slow grade system clock of 147,46 MHz, will downgrade the performance of the system.

4.4.2. The AD1937 requirements

The master clock source of the AD1937 should be between 6,9 MHz and 13,8 MHz, when operating from a 256 f_s reference, or at maximum 27,6 MHz when operating at direct 512 f_s .

“The on-chip PLL can be selected to reference the input sample rate from either of the LRCLK pins or 256 \times , 384 \times , 512 \times , or 768 \times sample rates (f_s), referenced to the 48 kHz mode from the MCLKI/MCLKXI pin. The default at power-up is 256 \times f_s from the MCLKI/MCLKXI pin. In 96 kHz mode, the master clock frequency stays at the same absolute frequency; therefore, the actual multiplication rate is divided by 2. In 192 kHz mode, the actual multiplication rate is divided by 4. For example, if the AD1937 is programmed in 256 \times f_s mode, the frequency of the master clock input is 256 \times 48kHz = 12.288MHz. If the AD1937 is then switched to 96 kHz operation (by writing to the I²C port), the frequency of the master clock should remain at 12,29 MHz, which is 128 \times f_s in this example. In 192 kHz mode, this becomes 64 \times f_s .

The internal clock for the ADCs is 256 \times f_s for all clock modes. The internal clock for the DACs varies by mode: 512 \times f_s (48 kHz mode), 256 \times f_s (96 kHz mode), or 128 f_s (192 kHz mode). By default, the on-board PLL generates this internal master clock from an external clock...

Note that it is not possible to use a direct clock for the ADCs set to the 192 kHz mode. It is required that the on-chip PLL be used in this mode.” [131]

What this means, is that if we were to use the CODEC in standalone mode, we would not be able to change the settings of the clock. This is not a problem, because the ADAU1467 can output a MCLK different than its input MCLK. This is a huge benefit, because we are not limited to use a 12,29 MHz master clock, and other options are available.

In conclusion, we will be using the higher 12,29 MHz because our board will not be impedance controlled and could

4.5. How we routed the CODEC to the DSP

To wire the AD1937 to the ADAU1467, the most straightforward connection would be to use one SDATA_IN port plus one SDATAIO pin, for the digital output of the ADCs, and one SDATA_OUT port plus three SDATAIO pins, for the digital input of the DACs. In this configuration, each data line would carry two channels in I²S mode. Almost no setup would be required in SigmaStudio, other than changing the SDATAIO pin routing to the serial data ports.

We will be using the standard stereo mode for the default routing, but because this wiring does not allow to access the other modes, we have decided to include a standard 2,54 mm header that splits the connection. The routing can be adjusted by the user by changing wiring of the header. Since each pair of adjacent pins serves as a jumper, the AD1937 can be completely disconnected.

You can see in the schematic in the figure on page 52 the wiring of the most common configurations. The benefit of this design is that the same header can be used to develop an expansion board. A secondary AD1937 in TDM mode would make this possible.

4.6. Characteristics of the DSP Development Board

Here is a description with some of the characteristics of the DSP development board:

- Power supply voltage: 5 V (DC)
- Power supply current: 1 A
- Number of analog inputs from CODEC: 2 stereo 3,5 mm TRS phono connectors
- Number of analog outputs from CODEC: 4 stereo 3,5 mm TRS phono connectors³
- Word/sample size of DSP: 32 bits
- Maximum sample rate: 192 kHz
- Minimum sample rate: 8 kHz
- Absolute maximum number of channels DSP: 48
- SDATA_IN/OUT ports available and channels used with the CODEC disabled: 4/4 (0 ch.)
- SDATA_IN/OUT ports available and channels used with the CODEC enabled: 3/3 (12 ch. in stereo mode, 24 ch. in AUX/TDM mode)
- SDATAIO pins available with the CODEC disabled: 8
- SDATAIO pins available with the CODEC in stereo mode: 4
- SDATAIO pins available with the CODEC in AUX/TDM mode: 8 or 7⁴
- Number of auxiliary ADC inputs: 10 (10-bit)
- Number of multipurpose pins (MPx) with CODEC disabled: up to 22⁵
- Number of multipurpose pins (MPx) with CODEC in stereo mode: up to 16
- Number of multipurpose pins (MPx) with CODEC in AUX/TDM mode: up to 20

³All the stereo inputs can be bypassed with the press of a switch to two of the output phono connectors.

⁴One SDATAIO pin is used in dual-line mode to achieve sample-rate of up to 192 kHz.

⁵Four MP pins always reserved for selfboot serial EEPROM.

- Extra peripherals: a manual reset push button, a thermal diode, two buffered clocks, an external reset pin, an internal power-down source, and a DSP SPI/I²C master control port.

The master control port shared the SPI lines with the serial EEPROM, but the I²S lines can be used in parallel to the EEPROM to control the on-board CODEC and any external devices.

4.7. How the DSP Development board should look like

After having designed the schematic, we chose the footprints for the components and added 3D models to it. The board layout was then developed in PCBNew, KiCADs layout design tool. We were not able to finish the design in the scheduled time, but the board was able to have an aspect similar looking to what we were expecting.

4.7.1. 3D View

Click on the image to open the view the 3D object. This requires a PDF reader with compatibility for version 1.7 or greater, like Adobe Reader.

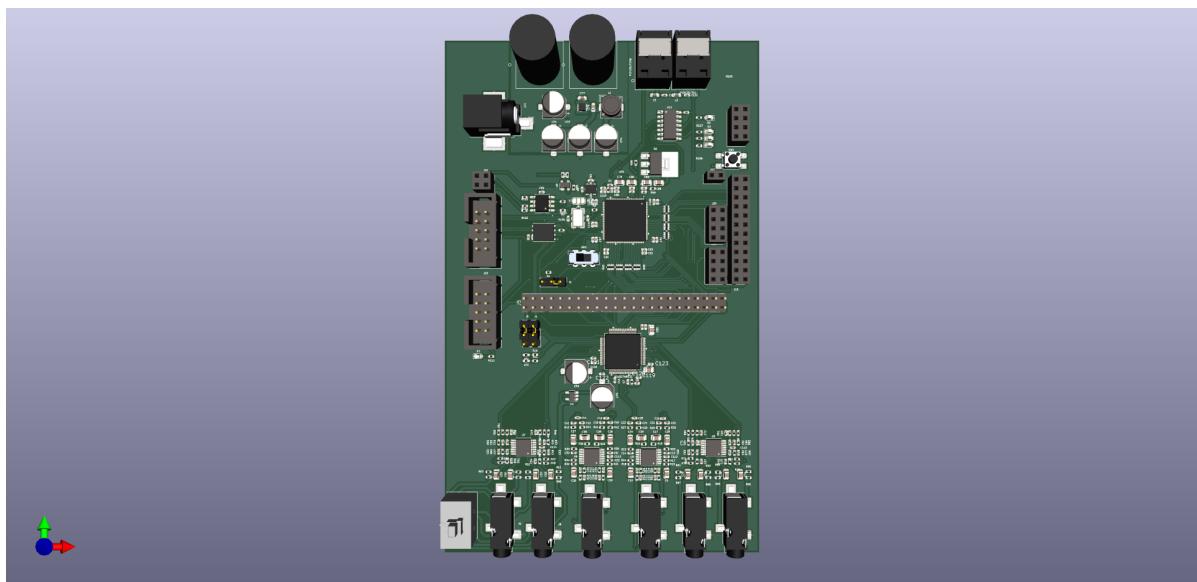


Figure 4.4.: U3D object of the DSP Development Platform

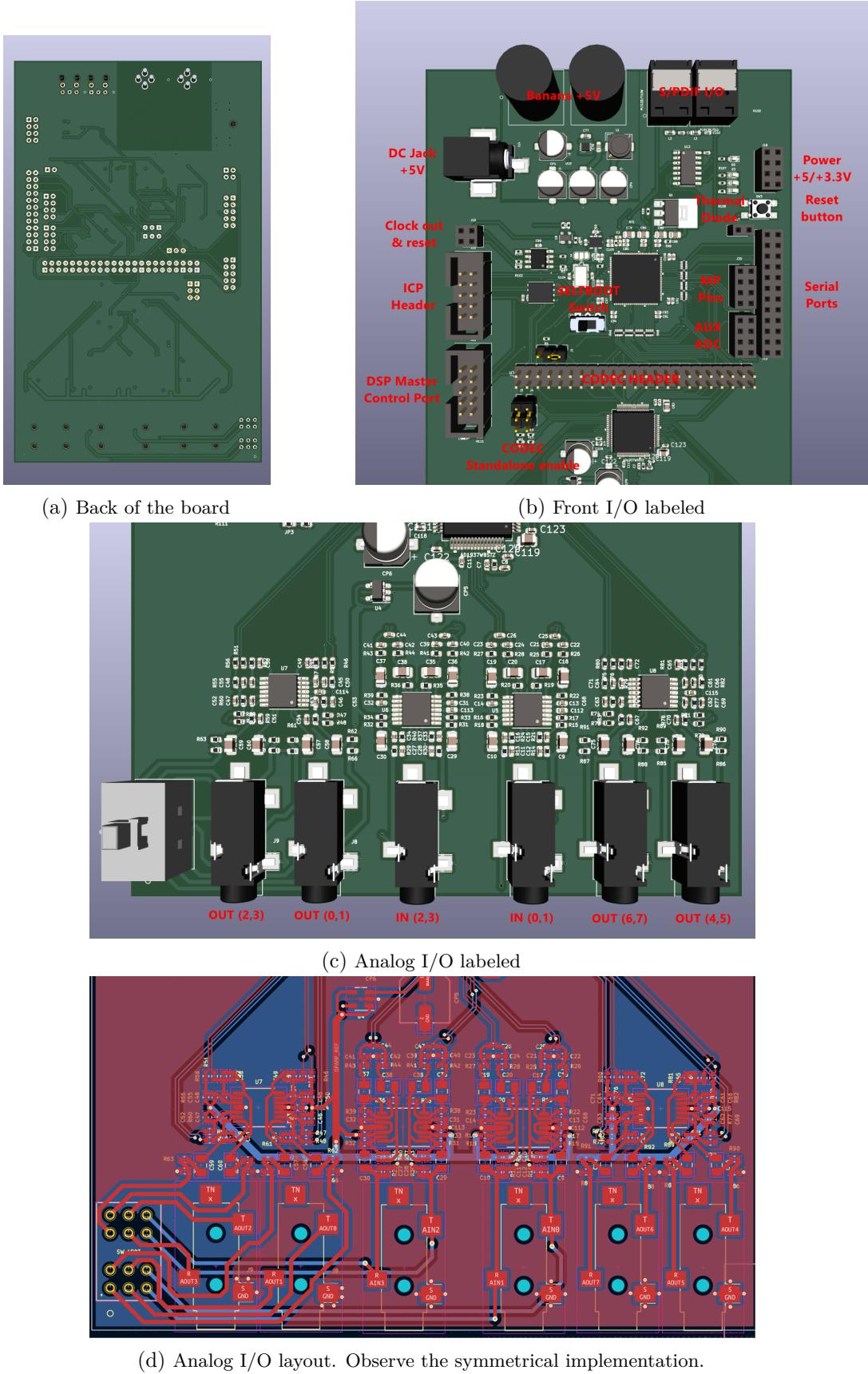


Figure 4.5.: Other views of the DSP Development Platform

4.8. Schematics of the DSP Development Board

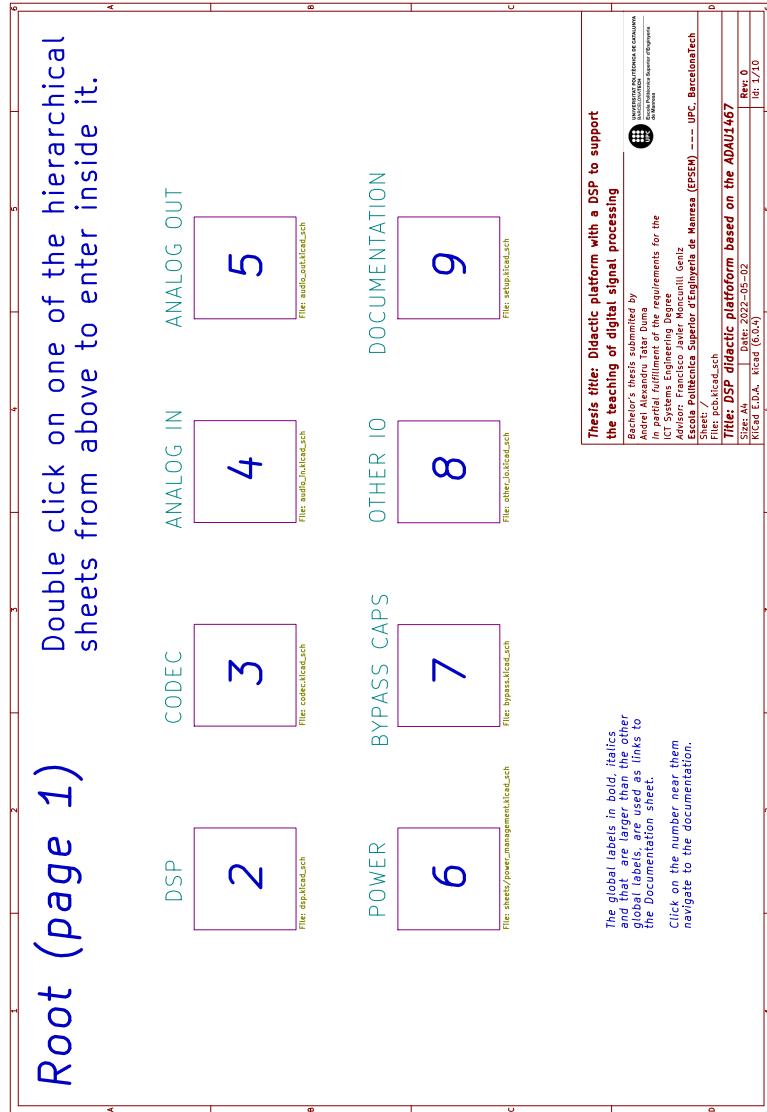
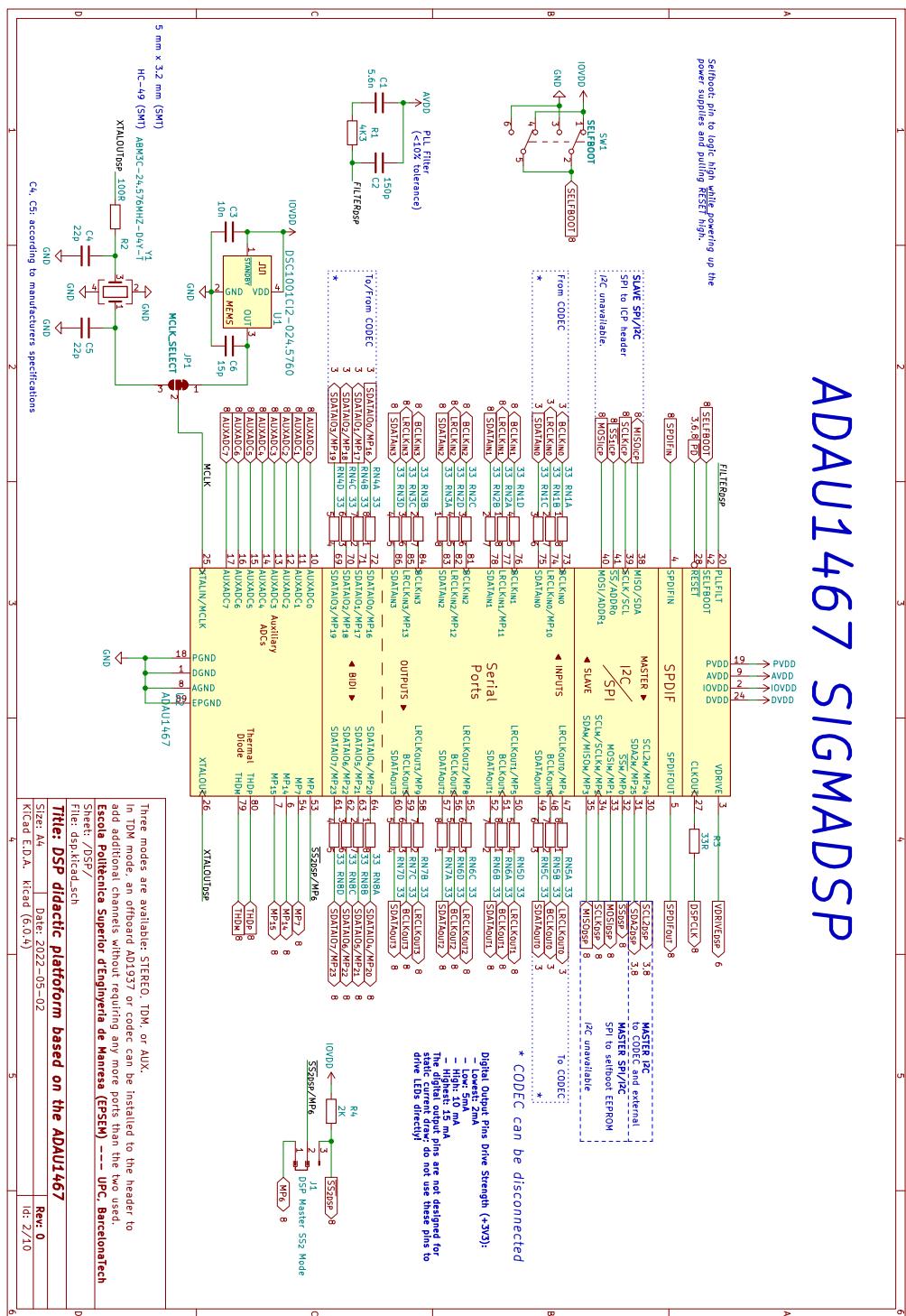


Figure 4.6.: DSP Development Platform schematic (page 1)

Page 1 of the schematic lacks any meaningful information for this document and its only purpose is for the developer to find his way to the different schematic sheets inside KiCAD. As you can see, each square has a number representing the page number of that sheet, and inside the EDA software, double-clicking it would navigate inside the sheet.

ADAU1467 SIGMADSP



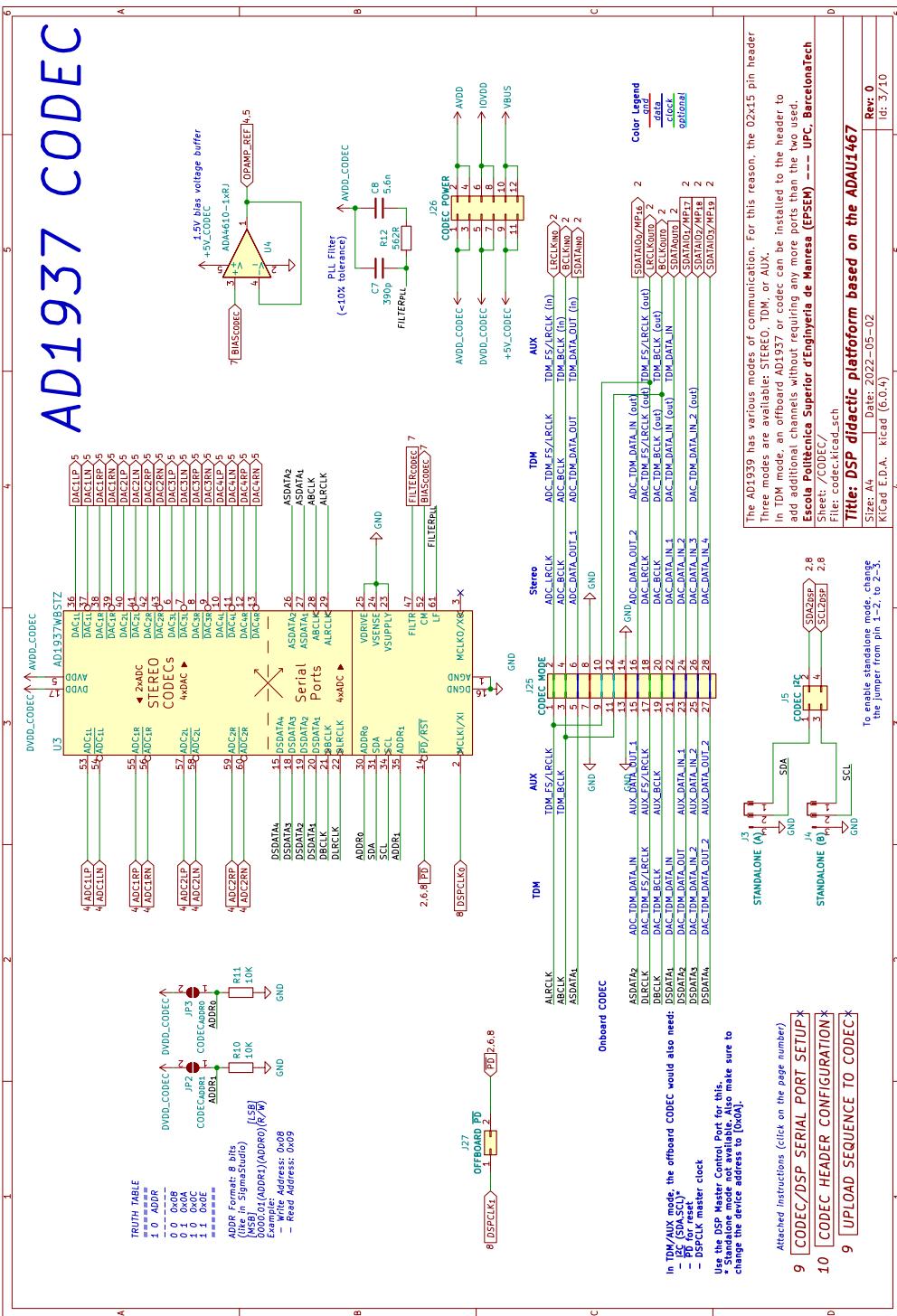
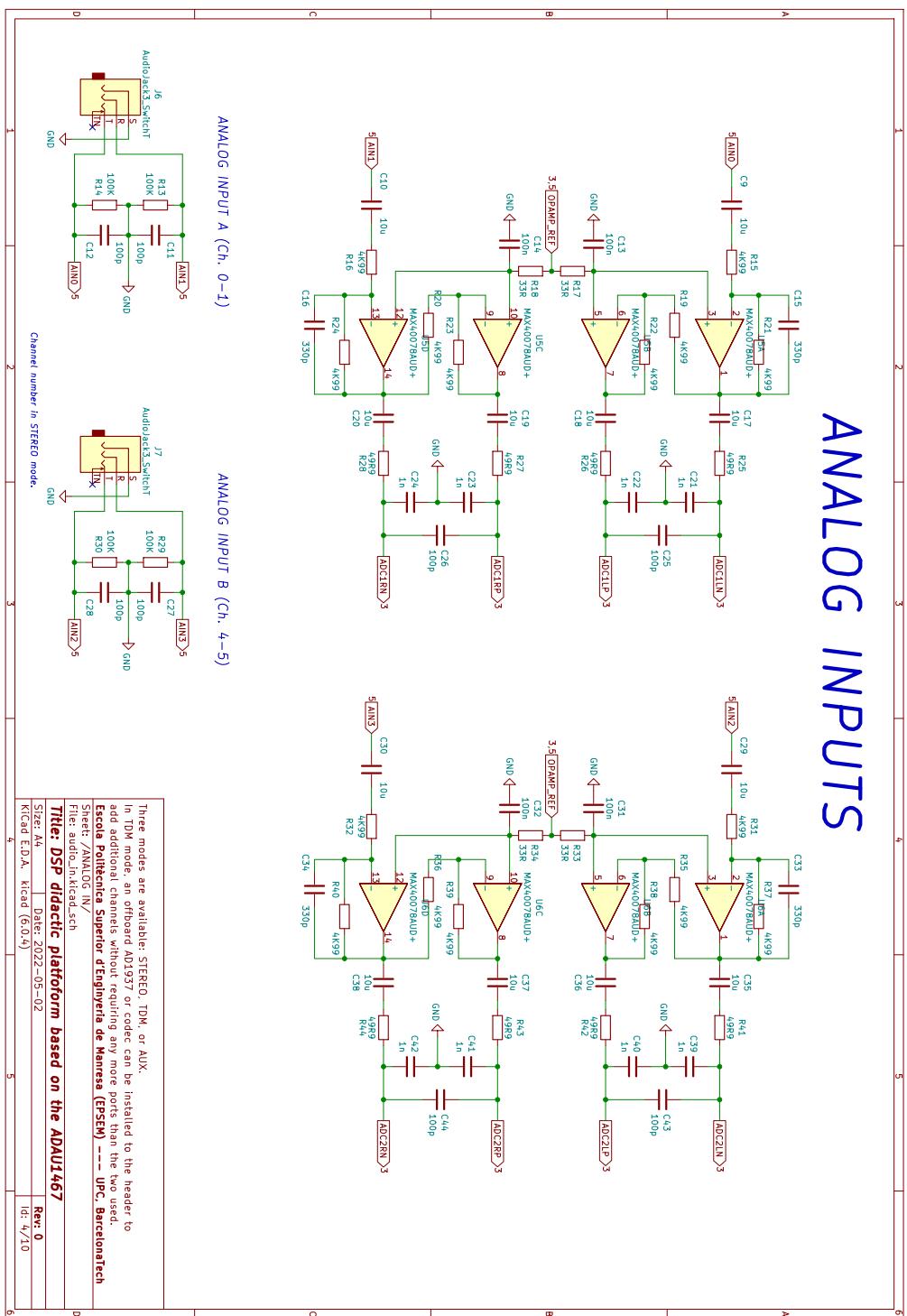


Figure 4.8.: DSP Development Platform schematic (page 3)



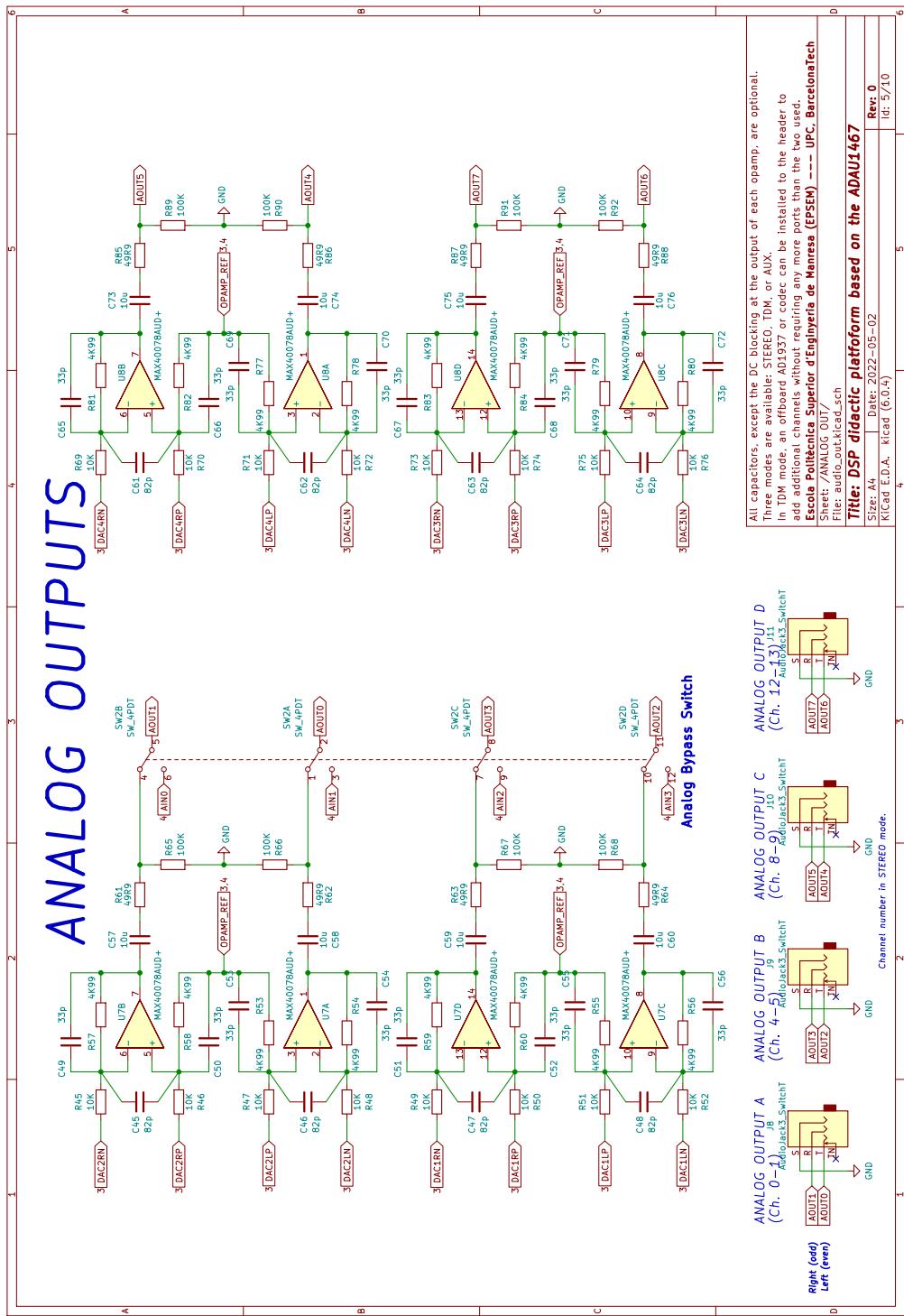


Figure 4.10.: DSP Development Platform schematic (page 5)

POWER SUPPLIES & SUPERVISION

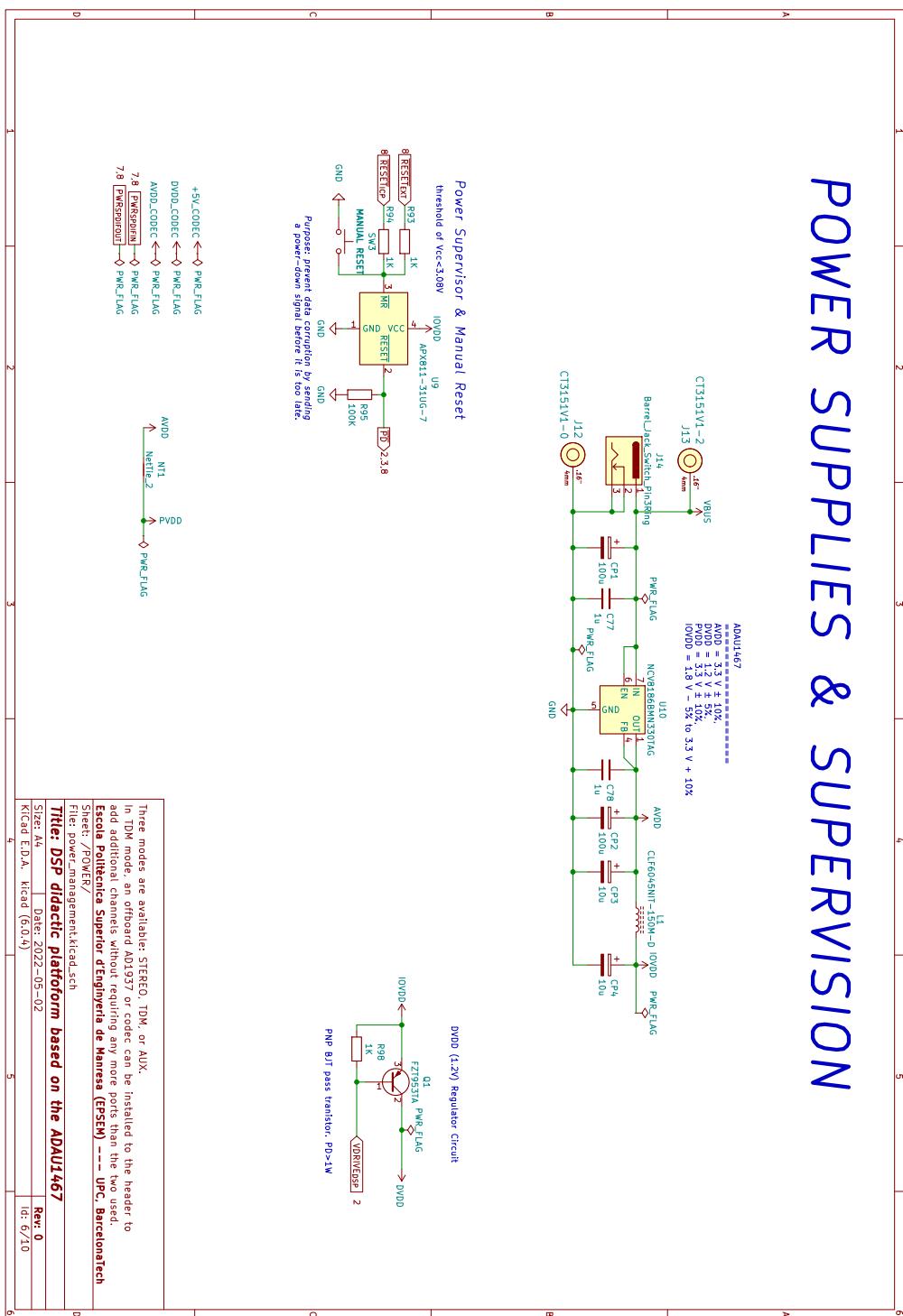


Figure 4.11.: DSP Development Platform schematic (page 6)

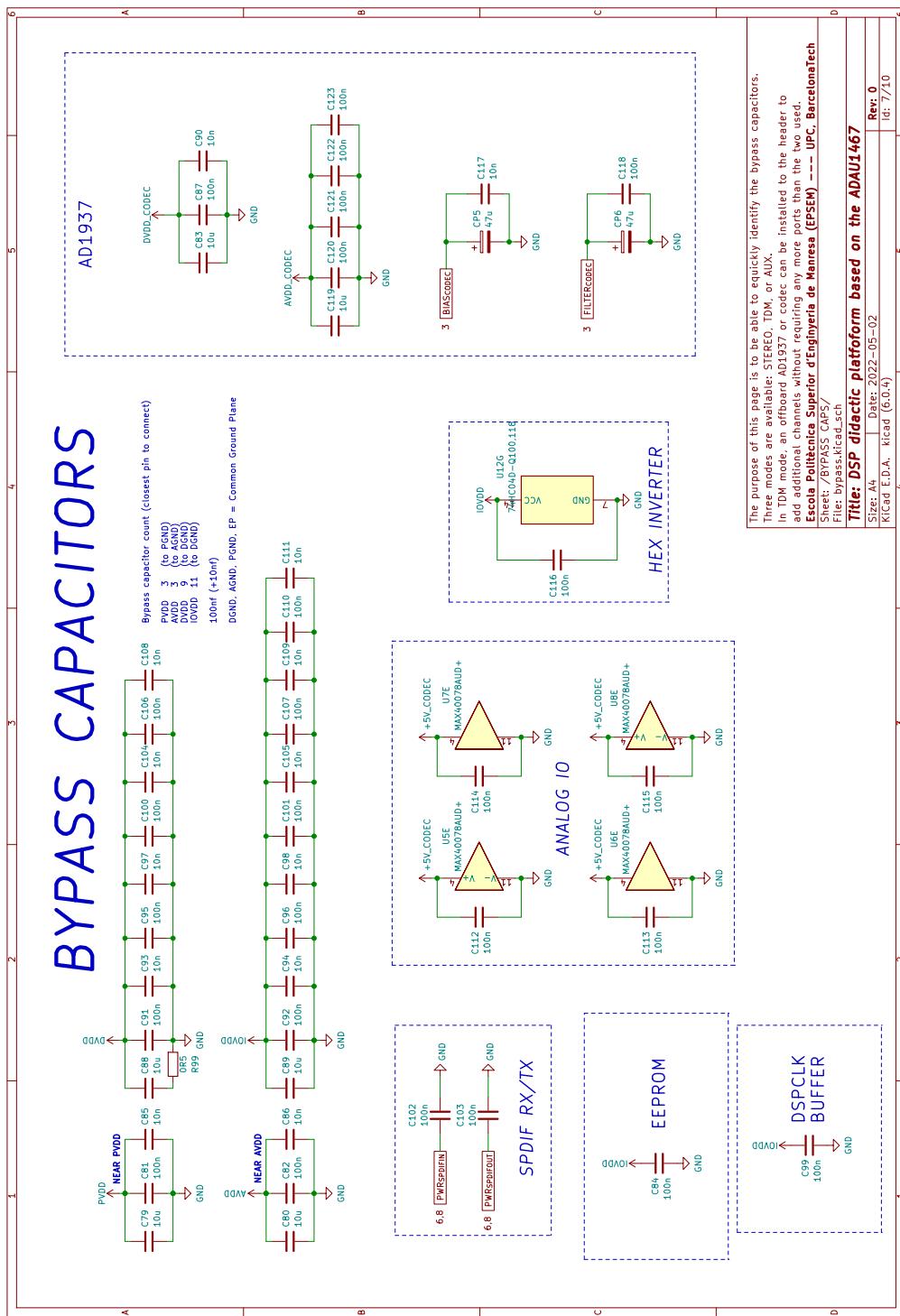


Figure 4.12.: DSP Development Platform schematic (page 7)

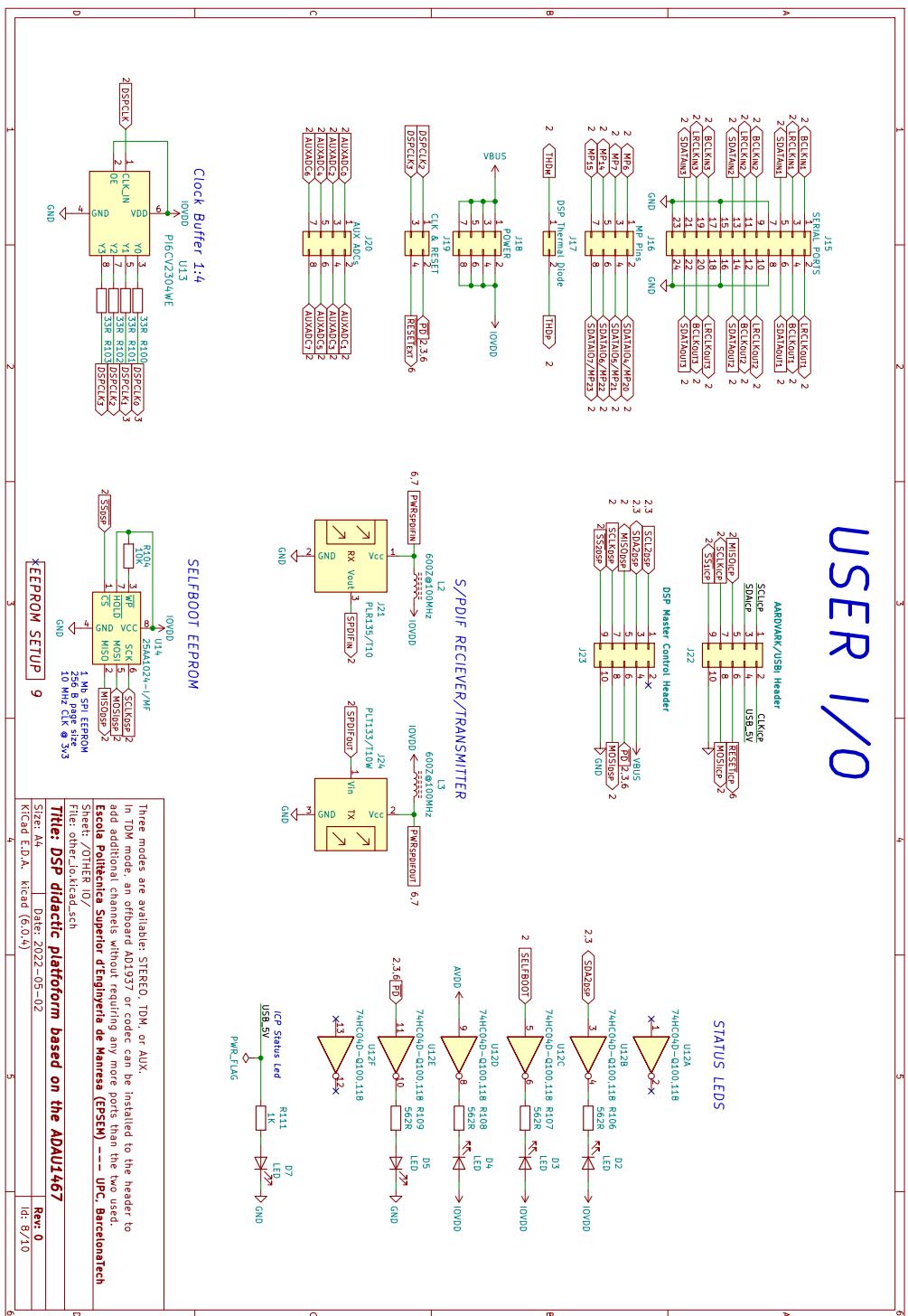


Figure 4.13.: DSP Development Platform schematic (page 8)

DOCUMENTATION

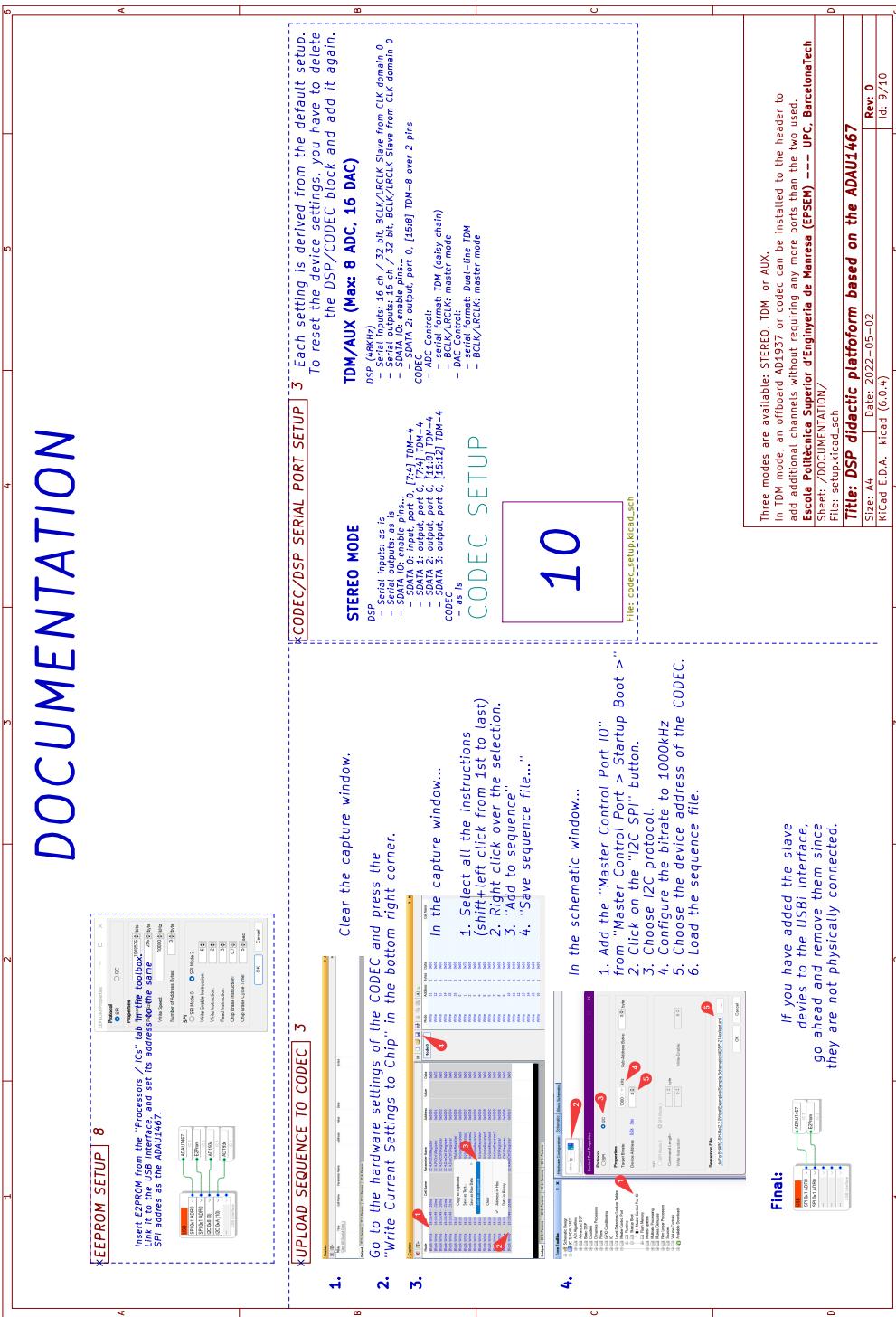


Figure 4.14.: DSP Development Platform schematic (page 9)

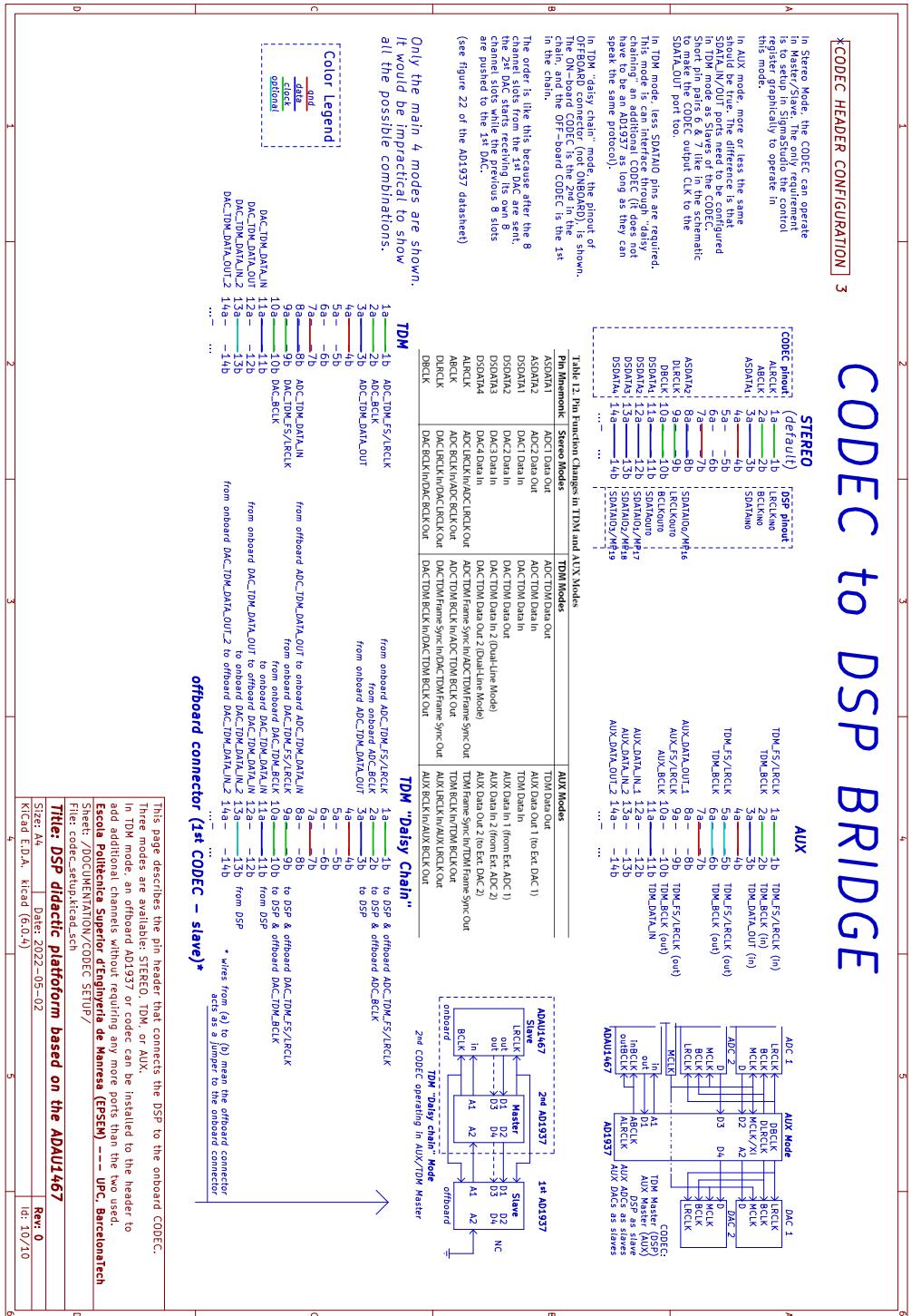


Figure 4.15.: DSP Development Platform schematic (page 10)

5. Conclusions & Future Work

In this thesis, we gave the reader the context in which we have based our thesis on, and decided on the main objective of it. Afterwards, we explained the students available materials with the end not to extend this list, *e.g.* by adding unavailable connectors.

Then, we have researched information about various didactic platforms that different manufacturers had to offer, so we could improve our understanding about these systems. While explaining the benefits of using graphical integrated development system, we compared different compatible products with SigmaStudio software, and chose an adequate digital signal processor in which to base our development platform on.

Before we began designing the PCB, we investigated about the common way of converting digital signals to analog, and vice versa, so we could add analog inputs to the DSP Development Platform.

Last but not least, we began the project of developing the schematic and explained some of the design choices.

One important lesson from this thesis was the time lost because of a critical error we made during our design decisions. Initially, when we were comparing different DSPs from Analog Devices, we ended up choosing the ADAU1787. In context, this device was tempting to use because it featured four in-chip ADCs & two DACs that with a headphone amplifier built-in. The best part was that the maximum sample-rate of this device is of 768 kHz. Similarly to the ADAU1467 we ended using, two 16-channel serial audio ports supporting I²S and up to TDM16 were available.

The problem we had by choosing this DSP is that it is packaged in a 42-ball, 0,35 mm pitch, 2,70 mm by 2,32 mm wafer-level chip-scale package (WLCSP) [132] BGA chip. The meaning of all this is that this device would have been extremely hard to handle, and even more to design around. PCBWay, a reputable PCB prototyping service, has quoted us upwards of 400€ for prototyping 5 boards with 4 layers given the advanced manufacturing capabilities [133].

In conclusion, we believe our thesis has achieved our desired ending point with our limited resources, and the objectives have been fulfilled to the best of our efforts given the aforementioned delay, with a finished design at the schematic and PCB levels, ready to be manufactured.

5.1. Future work

There are several additions that we would have liked to add to this project.

For starters, the DSP-to-CODEC bridge was a great feature our board has and the evaluation board Analog Devices presents do not. To be able to fully take advantage of this connector, a daughterboard compatible with the AUX/TDM modes of the AD1937 should propose a good expansion point for the system. Implementing this would not be trivial even with the help provided in the last sheet of the schematic, which breaks down the different modes into different pinouts. Hardware-wise, the optimized layout we proposed for the inputs and outputs can be copied over to the daughterboard to ease the designers work.

On a second note, the layout of the PCB should not be used as is because it requires a lot of touchup and improved routing of the different peripherals, as it was implemented rather quickly. We wanted to get at least an illustrative look of the board so the reader can observe how the layout should roughly look like in a hypothetical final design. For this reason, finishing the layout would be necessary.

Finally, if this thesis was ever to be used to design the DSP Development Platform in physical form, an accompanying laboratory practice for the students and a guide for the teachers and students should be developed.

One example that would require little audio processing, is to implement a way using incremental rotary encoders to change the volume of the stereo channels. Afterwards, we would try to save the data when a power-down happens, since this type of rotary encoders does not store the position. When the students accomplish this task, they could then implement a 3-band tone control (bass, middle and treble frequencies) using just a single rotary encoder and a button to change the attenuation/amplification on each band (using a state machine). What we had in mind with this example, is that using a graphical prototyping environment, the task should not take long to be finished.

Bibliography

- [1] James Morra. "Nexperia Begins Life Divorced from NXP Semiconductors". In: (Feb. 8, 2017). URL: <https://www.electronicdesign.com/power-management/article/21802257/nexperia-begins-life-divorced-from-nxp-semiconductors> (visited on 4/19/2022) (Cited on page x).
- [2] OEGlobal. *Universitat Politècnica de Catalunya. BarcelonaTech (UPC). Member Information.* English. 2022. URL: https://www.oeglobal.org/members/view/?member_id=550 (visited on 4/14/2022) (Cited on page 3).
- [3] Escola Politècnica Superior d'Enginyeria de Manresa. EPSEM – UPC. *Grau en Enginyeria de Sistemes TIC.* Catalan. 2022. URL: <https://www.epsem.upc.edu/ca/estudis/graus/grau-en-enginyeria-de-sistemes-tic> (visited on 4/14/2022) (Cited on page 3).
- [4] OpenCourseWare iTIC - UPC. *pds.* Catalan. Jan. 29, 2020. URL: <http://ocwitic.epsem.upc.edu/assignatures/pds> (visited on 4/13/2022) (Cited on page 5).
- [5] Stephen A. Dyer and Brian K. Harms. *Advances in Computers. Digital Signal Processing.* Ed. by Marchall C. Yovits. 1st ed. Vol. 37. London: Academic Press, Aug. 13, 1993, pp. 59–118. ISBN: 978-0120121373 (Cited on page 5).
- [6] Lizhe Tan and Jean Jiang. *Digital Signal Processing. Fundamentals and Applications.* Ed. by Steve Merken. Ed. by Jennifer Pierce. Ed. by Susan Ikeda. Ed. by Sruthi Satheesh. 3rd ed. London: Katey Birtcher, 2019. ISBN: 978-0-12-815071-9 (Cited on pages 5, 7).
- [7] Cem Cebenoyan. "GPU Gems. Chapter 28. Graphics Pipeline Performance". In: (2004). URL: <https://developer.nvidia.com/gpugems/gpugems/part-v-performance-and-practicalities/chapter-28-graphics-pipeline-performance> (visited on 4/19/2022) (Cited on page 5).
- [8] Wikipedia the free encyclopedia. *Digital signal processor.* Apr. 20, 2022. URL: https://en.wikipedia.org/wiki/Digital_signal_processor (visited on 4/20/2022) (Cited on page 5).
- [9] Texas Instruments. *Analog / Embedded processing / Semiconductor company.* 2022. URL: <https://www.ti.com/> (visited on 4/19/2022) (Cited on page 6).
- [10] Analog Devices. *Mixed-signal and digital signal processing ICs. Ahead of what's possible.* 2022. URL: <https://www.analog.com/> (visited on 4/19/2022) (Cited on page 6).
- [11] Next eXPerience Semiconductors. *NXP Semiconductors Official Site.* 2022. URL: <https://www.nxp.com/> (visited on 4/19/2022) (Cited on page 6).
- [12] ARM. *Artificial Intelligence Enhanced Computing.* 2022. URL: <https://www.arm.com/> (visited on 4/19/2022) (Cited on page 6).

- [13] Intel. *Intel® FPGAs and Programmable Devices*. 2022. URL: <https://www.intel.com/content/www/us/en/products/programmable.html> (visited on 4/19/2022) (Cited on page 6).
- [14] Advanced Micro Devices. *High-Performance & Adaptive Computing*. 2022. URL: <https://www.amd.com/> (visited on 4/19/2022) (Cited on page 6).
- [15] Advanced Micro Devices. *Xilinx. Adaptable. Intelligent.* 2022. URL: <https://www.xilinx.com/> (visited on 4/19/2022) (Cited on page 6).
- [16] Inc. Qualcomm Technologies. *Wireless Technology & Innovation / Mobile Technology*. 2022. URL: <https://www.qualcomm.com/> (visited on 4/19/2022) (Cited on page 6).
- [17] XMOS. *Bringing technology to life*. 2022. URL: <https://www.xmos.ai/> (visited on 4/26/2022) (Cited on page 6).
- [18] Inc. Cirrus Logic. *Cirrus Logic. Experts in Low Power, Mixed-Signal Processing*. 2022. URL: <https://www.cirrus.com/> (visited on 4/19/2022) (Cited on page 6).
- [19] Microchip Technology. *dsPIC33 Digital Signal Controllers and PIC24 MCUs*. 2022. URL: <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/16-bit-mcus> (visited on 4/20/2022) (Cited on pages 6, 11).
- [20] Wikipedia the free encyclopedia. *AVR microcontrollers*. Apr. 14, 2022. URL: https://en.wikipedia.org/wiki/AVR_microcontrollers (visited on 4/20/2022) (Cited on page 6).
- [21] Microsoft et al. *Visual Studio Code*. English. 2022. URL: <https://code.visualstudio.com/> (visited on 4/19/2022) (Cited on page 6).
- [22] James Yu. *Visual Studio Code LaTeX Workshop Extension*. English. 2022. URL: <https://github.com/James-Yu/LaTeX-Workshop> (visited on 4/19/2022) (Cited on page 6).
- [23] TeX Users Group. *T_EX Live*. Apr. 28, 2022. URL: <https://tug.org/texlive/> (visited on 5/21/2022) (Cited on page 6).
- [24] Jean-Pierre Charras et al. *Schematic Capture & PCB Design Software*. English. 2022. URL: <https://www.kicad.org/> (visited on 4/19/2022) (Cited on pages 6, 33).
- [25] EMA Design Automation. *Ultra Librarian. Free Online PCB CAD Library*. English. 2022. URL: <https://www.ultralibrarian.com/> (visited on 5/2/2022) (Cited on pages 6, 33).
- [26] Mouser Electronics Spain. *Electronic Components Distributor*. 2022. URL: <https://www.mouser.es/> (visited on 6/11/2022) (Cited on page 6).
- [27] Analog Devices. *LTspice Simulator*. 2022. URL: <https://www.analog.com/en/design-center/design-tools-and-calculators/ltpsice-simulator.html> (visited on 5/8/2022) (Cited on page 6).
- [28] The FreeCAD Team. *FreeCAD. Your own 3D parametric modeler*. English. 2022. URL: <https://www.freecadweb.org/> (visited on 5/2/2022) (Cited on page 6).
- [29] MeshLab. *The open source system for processing and editing 3D triangular meshes*. 2022. URL: <https://www.meshlab.net/> (visited on 6/11/2022) (Cited on page 6).
- [30] Alexander Grahn. *Comprehensive T_EX Archive Network. Package media9*. Mar. 18, 2022. URL: <https://www.ctan.org/pkg/media9> (visited on 6/11/2022) (Cited on page 6).

- [31] Arduino. "What is Arduino?" In: (Feb. 5, 2018). URL: <https://www.arduino.cc/en/Guide/Introduction> (visited on 4/20/2022) (Cited on page 7).
- [32] Inc. The MathWorks. *Simulink. Simulation and Model-Based Design*. Apr. 20, 2022. URL: <https://www.mathworks.com/products/simulink.html> (visited on 4/20/2022) (Cited on page 7).
- [33] Shekhar Sharad and National Instruments. "Graphical programming for DSPs". In: (June 25, 2007). URL: <https://www.eetimes.com/graphical-programming-for-dsp/> (visited on 4/19/2022) (Cited on page 7).
- [34] National Instruments. *What is LabVIEW?* 2022. URL: <https://www.ni.com/en-us/shop/labview.html> (visited on 4/19/2022) (Cited on page 7).
- [35] DSP Robotics. *FlowStone Applications*. 2022. URL: <http://www.dsprobotics.com/applications.html> (visited on 4/19/2022) (Cited on page 7).
- [36] Inc. Terasic. *DE0-Nano Development and Education Board*. 2022. URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English%5C&No=593> (visited on 4/21/2022) (Cited on page 7).
- [37] Corp. Intel. *Cyclone IV EP4CE22 FPGA*. 2022. URL: <https://www.intel.com/content/www/us/en/products/sku/210468/cyclone-iv-ep4ce22-fpga/specifications.html> (visited on 4/21/2022) (Cited on page 7).
- [38] OpenCourseWare iTIC - UPC. *Sistemes Encastats (SE)*. Catalan. Nov. 21, 2017. URL: <http://ocwitic.epsem.upc.edu/assignatures/se> (visited on 4/13/2022) (Cited on page 7).
- [39] OpenCourseWare iTIC - UPC. *Material per a les pràctiques d'Introducció als Sistemes Digitals*. Catalan. 2017. URL: <http://ocwitic.epsem.upc.edu/assignatures/isd/practiques/material-per-a-les-practiques-dintroduccio-als-sistemes-digitals/view> (visited on 4/20/2022) (Cited on page 8).
- [40] Keysight Technologies. *U8903B Performance Audio Spectrum Analyzer / Keysight*. 2022. URL: <https://www.keysight.com/es/en/product/U8903B/performance-audio-analyzer.html> (visited on 4/20/2022) (Cited on page 8).
- [41] Wikipedia the free encyclopedia. *Modular synthesizer*. Apr. 7, 2022. URL: https://en.wikipedia.org/wiki/Modular_synthesizer (visited on 4/20/2022) (Cited on page 8).
- [42] JEITA. *EIAJ RC-5325A. 4-Pole miniature concentric plugs and jacks*. English and Japanese. 2010. URL: <https://www.jeita.or.jp/japanese/standard/book/RC-5325A> (visited on 5/6/2022) (Cited on page 9).
- [43] JEITA. *Japan Electronics and Information Technology Industries Association*. Japanese. 2022. URL: <https://www.jeita.or.jp/> (visited on 5/6/2022) (Cited on page 9).
- [44] CTIA. *Cellular Telecommunications and Internet Association. The Wireless Association*. 2021. URL: <https://www.ctia.org/> (visited on 5/6/2022) (Cited on page 9).
- [45] NIPPON DICS. *Pentaconn*. 2022. URL: <https://ndics.com/en/products/pentaconn/> (visited on 5/12/2022) (Cited on page 9).
- [46] Head-Fi.org. *NEW Sony 4.4mm Pentaconn Balanced Connector. Headphone Reviews and Discussion*. Ed. by Moon Audio. Jan. 27, 2017. URL: <https://www.head-fi.org/threads/new-sony-4-4mm-pentaconn-balanced-connector.833555/> (visited on 5/12/2022) (Cited on page 9).

- [47] Head-Fi.org. *3.5mm and 2.5mm TRRS balanced cable questions... Headphone Reviews and Discussion*. Ed. by nraymond. Jan. 10, 2018. URL: <https://www.head-fi.org/threads/3-5mm-and-2-5mm-trrs-balanced-cable-questions.869527/> (visited on 5/12/2022) (Cited on page 10).
- [48] basic knowledge of communication terms. *JEITA RC-8141C*. Japanese. 2022. URL: <https://www.wdic.org/w/WDIC/RC-8141C> (visited on 5/12/2022) (Cited on page 10).
- [49] Wikipedia the free encyclopedia. *Sony/Philips Digital Interface*. June 6, 2022. URL: <https://en.wikipedia.org/wiki/S/PDIF> (visited on 6/12/2022) (Cited on page 10).
- [50] ARM. *Education Kits*. English. 2022. URL: <https://www.arm.com/resources/education/education-kits> (visited on 4/24/2022) (Cited on page 11).
- [51] ARM. *ARM Education*. English. 2022. URL: <https://github.com/arm-university> (visited on 4/24/2022) (Cited on page 11).
- [52] STMicroelectronics. *STM32F4DISCOVERY. Discovery kit with STM32F407VG MCU*. English. 2022. URL: <https://www.st.com/en/evaluation-tools/stm32f4discovery.html> (visited on 4/24/2022) (Cited on page 11).
- [53] STMicroelectronics. *STM32F4DISCOVERY. Discovery kit with STM32F407VG MCU*. English. 2022. URL: <https://www.st.com/en/evaluation-tools/32f746gdiscovery.html> (visited on 4/24/2022) (Cited on page 11).
- [54] Cem Ünsalan, M. Erkin Yücel, and H. Deniz Gürhan. *Digital Signal Processing using Arm Cortex-M based Microcontrollers. Theory and Practice*. epub. Cambridge: Arm Education Media, 2018. Chap. 1. ISBN: 978-1-911531-15-9. URL: <https://www.arm.com/resources/ebook/digital-signal-processing> (visited on 4/24/2022) (Cited on page 12).
- [55] Skyler Cornell. “Audio DSP Development Board”. Bryn Mawr College, 2021. URL: <http://hdl.handle.net/10066/23084> (visited on 4/24/2022) (Cited on page 11).
- [56] Cirrus Logic. *CS5343/44. 98 dB, 96 kHz, Low Power Stereo A/D Converters*. English. 2022. URL: <https://www.cirrus.com/products/cs5343-44/> (visited on 4/14/2022) (Cited on page 11).
- [57] Cirrus Logic. *CS4344/45/48. 10-pin, 24-bit, 192 kHz Stereo D/A Converters*. English. 2022. URL: <https://www.cirrus.com/products/cs4344-45-48/> (visited on 4/14/2022) (Cited on page 11).
- [58] Wikipedia the free encyclopedia. *digital signal controller*. Jan. 13, 2022. URL: https://en.wikipedia.org/wiki/Digital_signal_controller (visited on 4/25/2022) (Cited on page 11).
- [59] Microchip Technology. *PIC32 Digital Audio*. 2022. URL: <https://www.microchip.com/en-us/solutions/consumer/audio-and-speech/pic32-digital-audio> (visited on 4/25/2022) (Cited on page 11).
- [60] Microchip Technology. *EXPLORER 16/32 DEVELOPMENT KIT. Part Number: DM240001-3*. 2022. URL: <https://www.microchip.com/en-us/development-tool/dm240001-3> (visited on 4/25/2022) (Cited on page 11).
- [61] MikroElektronika. *mikroBUS*. 2022. URL: <https://www.mikroe.com/mikrobus> (visited on 4/25/2022) (Cited on page 11).

- [62] Analog Devices. *EngineerZone*. 2022. URL: <https://ez.analog.com/> (visited on 4/26/2022) (Cited on page 13).
- [63] Analog Devices. *Education Engagement. License to CrossCore Embedded Studio*. 2022. URL: <https://www.analog.com/en/education/university-engagement.html> (visited on 4/28/2022) (Cited on pages 13, 17).
- [64] Analog Devices. *Evaluation Hardware & Software. Design Center*. 2022. URL: <https://www.analog.com/en/design-center/evaluation-hardware-and-software.html> (visited on 4/26/2022) (Cited on page 13).
- [65] Analog Devices. *Audio Signal Processors. Audio and Video Products*. 2022. URL: <https://www.analog.com/en/product-category/audio-signal-processors.html> (visited on 4/26/2022) (Cited on page 13).
- [66] Wikipedia the free encyclopedia. *Super Harvard Architecture Single-Chip Computer*. Nov. 1, 2021. URL: https://en.wikipedia.org/wiki/Super_Harvard_Architecture_Single-Chip_Computer (visited on 4/20/2022) (Cited on page 13).
- [67] Analog Devices. *EVAL-ADAU1701MINIZ. ADAU1701 Mini Evaluation Board*. 2022. URL: <https://www.analog.com/media/en/technical-documentation/evaluation-documentation/EVAL-ADAU1701MINIZ.pdf> (visited on 4/26/2022) (Cited on page 13).
- [68] Digi-Key Electronics. *EVAL-ADAU1701MINIZ Analog Devices Inc. Development Boards, Kits, Programmers*. 2022. URL: <https://www.digikey.ca/en/products/detail/analog-devices-inc/EVAL-ADAU1701MINIZ/1857436> (visited on 4/26/2022) (Cited on page 15).
- [69] miniDSP. *Active Crossovers: miniDSP 2x4*. 2022. URL: <https://www.minidsp.com/products/minidsp-in-a-box/minidsp-2x4> (visited on 4/26/2022) (Cited on page 13).
- [70] miniDSP. *miniDSP Kits: miniDSP 2x4 kit*. 2022. URL: <https://www.minidsp.com/products/minidspkits/2-x-in-4-x-out> (visited on 4/26/2022) (Cited on page 13).
- [71] miniDSP. *miniDSP Kits: miniDSP 2x4 HD kit*. 2022. URL: <https://www.minidsp.com/products/minidspkits/2-x-in-4-x-out-hd> (visited on 4/26/2022) (Cited on page 14).
- [72] XMOS. *xCORE-200 USB multicore microcontrollers*. 2019. URL: <https://origin.xmos.com/developer/silicon/xcore200-usb> (visited on 4/26/2022) (Cited on page 14).
- [73] Analog Devices. “VisualDSP++ 5.1”. In: (2022). URL: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/software/vdsp-bf-sh-ts.html#software-overview> (visited on 5/22/2022) (Cited on page 14).
- [74] Theo Verelst. *Local Diary Page 50*. July 21, 2008. URL: <http://www.theover.org/Diary/ldiary50.html> (visited on 5/22/2022) (Cited on page 16).
- [75] Analog Devices. *CrossCore Embedded Studio*. 2022. URL: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/software/adswt-cces.html> (visited on 4/29/2022) (Cited on page 14).
- [76] Analog Devices. *SigmaStudio*. 2022. URL: https://www.analog.com/en/design-center/evaluation-hardware-and-software/software/ss_sigst_02.html (visited on 4/26/2022) (Cited on page 14).

- 20LFCSP%20(5mm%20x%205mm%20w/%20EP)%7C48-Lead%20LQFP%20(7mm%20x%207mm)%7C72-Lead%20LFCSP%20(10mm%20x%2010mm%20w/%20EP)%7C80-Lead%20LQFP%20(14mm%20x%2014mm)%7C88-Lead%20LFCSP%20(12mm%20x%2012mm%20w%200.85mm%20w/%20EP)&p5327=%7C4%7C8 (visited on 5/23/2022) (Cited on page 18).
- [88] Analog Devices. *ADAU1401A. SigmaDSP 28-/56-Bit Audio Processor with Two ADCs and Four DACs.* 2010. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1401A.pdf> (visited on 5/30/2022) (Cited on page 19).
- [89] Analog Devices. *ADAU1701. SigmaDSP 28-/56-Bit Audio Processor with Two ADCs and Four DACs.* 2016. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1701.pdf> (visited on 5/1/2022) (Cited on page 19).
- [90] Analog Devices. *ADAU1761. SigmaDSP Stereo, Low Power, 96 kHz, 24-Bit Audio Codec with Integrated PLL.* 2022. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1761.pdf> (visited on 5/1/2022) (Cited on page 19).
- [91] Analog Devices. *ADAU1442/ADAU1445/ADAU1446. SigmaDSP Digital Audio Processor with Flexible Audio Routing Matrix.* 2013. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1442_1445_1446.pdf (visited on 5/30/2022) (Cited on page 19).
- [92] NXP. “I²S bus specification. Archive from the Wayback Machine”. In: (June 5, 1996). URL: https://web.archive.org/web/20070102004400/http://www.nxp.com/acrobat_download/varioust/I2SBUS.pdf (visited on 6/2/2022) (Cited on page 21).
- [93] Analog Devices. *ADAU1463/ADAU1467. SigmaDSP Digital Audio Processor.* 2022. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADAU1463-1467.pdf> (visited on 5/2/2022) (Cited on page 21).
- [94] NXP. “UM10204 - I²C-bus specification and user manual. Archive from the Wayback Machine”. In: (Apr. 4, 2014). URL: <https://web.archive.org/web/20210813122132/https://www.nxp.com/docs/en/user-guide/UM10204.pdf> (visited on 6/2/2022) (Cited on page 21).
- [95] Analog Devices. *Selection Table for Audio CODECs. Parametric Search of: AD19.* 2022. URL: <https://www.analog.com/en/parametricsearch/11357#/sort=0,asc&p0=AD19> (visited on 5/30/2022) (Cited on page 22).
- [96] Analog Devices. *AD1939 Datasheet (Rev.E). 4 ADC/8 DAC with PLL, 192 kHz, 24-Bit CODEC.* 2013. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD1939.pdf> (visited on 4/29/2022) (Cited on page 25).
- [97] Analog Devices. *ADAU1467 booting 3 x AD1937 codec's. Q&A - Processors and DSP - EngineerZone.* Ed. by MDThommo81. Sept. 4, 2021. URL: <https://ez.analog.com/dsp/f/q-a/549309/adau1467-booting-3-x-ad1937-codec-s> (visited on 6/7/2022) (Cited on page 26).
- [98] Analog Devices Wiki. *Master Control Port Boot time I/O (ADAU145x) []*. Ed. by Joshua Berlin. Jan. 27, 2020. URL: <https://wiki.analog.com/resources/tools-software/sigmastudio/toolbox/mastercontrolport/mastercontrolportio> (visited on 6/7/2022) (Cited on page 26).

- [99] Wikipedia the free encyclopedia. *Differential Signalling*. May 3, 2022. URL: https://en.wikipedia.org/wiki/Differential_signalling (visited on 5/6/2022) (Cited on page 26).
- [100] Ethan Winer. *The Audio Expert: Everything You Need to Know About Audio*. Focal Press, 2013, p. 107. ISBN: 978-0-240-82100-9 (Cited on page 27).
- [101] Douglas Self. *Small signal audio design*. 3rd ed. New York: Routledge, 2020. ISBN: 978-1-003-03183-3 (Cited on page 27).
- [102] THAT Corporation. *InGenius High-CMRR Balanced Input Line Receiver ICs*. 2020. URL: http://www.thatcorp.com/datasheets/THAT_1200-Series_Datasheet.pdf (visited on 5/7/2022) (Cited on page 27).
- [103] Bill Whitlock. “Interconnection of Balanced and Unbalanced Equipment. Jensen AN-003”. In: (1995). URL: <https://www.jensen-transformers.com/wp-content/uploads/2014/08/an003.pdf> (visited on 5/7/2022) (Cited on page 27).
- [104] Edmund A. Laport. *Balanced to Unbalanced Transformations*. Mar. 19, 2011. URL: http://www.vias.org/radioanteng/radio_antenna_engineering_04_06.html (visited on 5/7/2022) (Cited on page 27).
- [105] Analog Devices. *Op Amp Applications Handbook*. Ed. by Walt Jung. 2005. URL: <https://www.analog.com/en/education/education-library/op-amp-applications-handbook.html> (visited on 5/10/2022) (Cited on page 27).
- [106] Bruce Carter. *A Single-Supply Op-Amp Circuit Collection. SLOA058 Application Report*. Nov. 2000. URL: http://www.ece.uah.edu/~jovanov/msp430/app_notes/SingleSupply_op-amps_SLOA058.pdf (visited on 5/11/2022) (Cited on page 29).
- [107] DiySong. *Diamond Buffer Exploration*. chinese. URL: <http://diysong.com/report/images/diamonbuffer/DiamondBuffer.htm> (visited on 5/20/2022) (Cited on page 29).
- [108] Maxim Integrated. *MAX9722A 5 V, Differential Input, DirectDrive, 130 mW Stereo Headphone Amplifiers with Shutdown*. 2022. URL: <https://www.maximintegrated.com/en/products/analog/audio/MAX9722A.html> (visited on 5/11/2022) (Cited on page 29).
- [109] Adrian Rolufs. *Overview of DirectDrive Technology. Tutorial 3979, AN3979*. May 8, 2007. URL: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/3/3979.html> (visited on 5/11/2022) (Cited on page 29).
- [110] Maxim Integrated. *MAX97220A Differential Input DirectDrive Line Drivers/Headphone Amplifiers*. 2022. URL: <https://www.maximintegrated.com/en/products/analog/audio/MAX97220A.html> (visited on 5/15/2022) (Cited on page 29).
- [111] Analog Devices. *EVAL-ADAU1467Z. UG-1134*. 2022. URL: <https://www.analog.com/media/en/technical-documentation/user-guides/EVAL-ADAU1467Z-UG-1134.pdf> (visited on 5/31/2022) (Cited on page 30).
- [112] Analog Devices. *ADAU1787 Evaluation Board*. 2022. URL: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/EVAL-ADAU1787Z.html> (visited on 5/2/2022) (Cited on page 29).
- [113] Texas Instruments. *CD4016B. CMOS Quad bilateral switch*. 2003. URL: <https://www.ti.com/lit/ds/symlink/cd4016b.pdf> (visited on 5/13/2022) (Cited on page 30).

- [114] Analog Devices. *ADG884. 0,5 Ω CMOS, Dual 2 : 1 MUX/SPDT Audio Switch*. 2012. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADG884.pdf> (visited on 5/13/2022) (Cited on page 30).
- [115] E-Switch. *TL2201 and TL4201 Series Subminiature Pushbutton Switches*. 2022. URL: <https://www.e-switch.com/product-catalog/pushbutton/product-lines/tl2201-and-tl4201-series-subminiature-pushbutton-switches> (visited on 5/13/2022) (Cited on page 30).
- [116] Stack Exchange - Electrical Engineering. *Toslink (fiber optic) connectors not working*. Ed. by user2665581. Ed. by Passerby. 2013-10-29. URL: <https://electronics.stackexchange.com/questions/83800/toslink-fiber-optic-connectors-not-working> (visited on 5/31/2022) (Cited on page 31).
- [117] Analog Devices. *EVAL-ADAU1467Z - writeback circuit purpose*. Ed. by MDThommo81. Aug. 8, 2021. URL: <https://ez.analog.com/dsp/sigmadsp/f/q-a/548230/eval-adau1467z---writeback-circuit-purpose> (visited on 6/9/2022) (Cited on page 35).
- [118] Arduino. *Flash USB-to-serial firmware in DFU mode*. *Arduino Help Center*. English. May 13, 2022. URL: <https://support.arduino.cc/en-us/articles/4408887452434-Flash-USB-to-serial-firmware-in-DFU-mode> (visited on 5/15/2022) (Cited on page 36).
- [119] Future Technology Devices International. *FTDI*. 2022. URL: <https://ftdichip.com/> (visited on 5/14/2022) (Cited on page 36).
- [120] Arduino. *FT232R USB UART IC Datasheet. Version 2.16*. May 21, 2020. URL: https://ftdichip.com/wp-content/uploads/2020/08/DS_FT232R.pdf (visited on 5/15/2022) (Cited on page 36).
- [121] Total Phase. *Aardvark I²C/SPI Host Adapter User Manual v5.15*. English. Feb. 28, 2014. URL: https://www.totalphase.com/media/support/article_attachments/200626813/aardvark-v5.15.pdf (visited on 5/15/2022) (Cited on page 36).
- [122] Infineon Technologies. *Cypress acquisition*. 2020. URL: <https://www.infineon.com/cms/en/about-infineon/company/cypress-acquisition/> (visited on 5/15/2022) (Cited on page 36).
- [123] Sure Electronics. *WONDOM ICP1 User Guide*. URL: <https://store.sure-electronics.com/images/documents/WONDOM%20ICP1%20User%20Guide.pdf> (visited on 5/15/2022) (Cited on page 36).
- [124] Infineon Technologies. *CY7C68013A-56PVXC. EZ-USB FX2LP*. Aug. 20, 2021. URL: <https://www.infineon.com/cms/en/product/universal-serial-bus-usb-power-delivery-controller/peripheral-controllers/ez-usb-fx2lp/cy7c68013a-56pvxc/> (visited on 5/15/2022) (Cited on page 36).
- [125] AliExpress. *EZ-USB FX2LP CY7C68013A USB logic analyzer core board+Source Code*. Ed. by Black Hole Tech Store. URL: <https://www.aliexpress.com/item/1005001659197850.html> (visited on 5/15/2022) (Cited on page 36).
- [126] instalator. “USB Interface for DSP Programming Analog Devices audio processors in Sigma Studio”. Russian. In: (May 24, 2019). URL: <https://blog.instalator.ru/archives/1560> (visited on 5/15/2022) (Cited on pages 36, 70).

- [127] Analog Devices. *Can ADI provide or support the USB interface firmware code for my self-assembled PCB or end product? SigmaDSP Processors and SigmaStudio Development Tool - EngineerZone*. URL: <https://ez.analog.com/dsp/sigmadsp/w/documents/5202/can-adi-provide-or-support-the-usb-interface-firmware-code-for-my-self-assembled-pcb-or-end-product> (visited on 5/16/2022) (Cited on page 36).
- [128] ChipDip. “SigmaLink-USBi, USBi programmer for SigmaStudio”. Russian. In: (2022). URL: <https://www.chipdip.ru/product/sigmalink-usbi> (visited on 5/15/2022) (Cited on page 36).
- [129] Texas Instruments. *Voltage Translation Between 3.3-V, 2.5-V, 1.8-V, and 1.5-V Logic Standards*. Apr. 2015. URL: https://www.ti.com/lit/an/scea030b/scea030b.pdf?ts=1654761553068&ref_url=https%253A%252Fwww.ti.com%252Fproduct%252FLSF0101 (visited on 6/9/2022) (Cited on page 37).
- [130] Texas Instruments. *LSF010x 1/2/8 Channel Auto-Bidirectional Multi-Voltage Level Translator for Open-Drain and Push-Pull Applications*. May 2021. URL: <https://www.ti.com/lit/ds/symlink/lsf0108.pdf> (visited on 6/10/2022) (Cited on page 37).
- [131] Analog Devices. *AD1937 Datasheet (Rev.B)*. 4 ADC/8 DAC with PLL, 192 kHz, 24-Bit CODEC. 2010. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD1937.pdf> (visited on 4/29/2022) (Cited on page 39).
- [132] Broadcom. *waffer-level chip-scale package application note. Overview and assembly guidelines*. Dec. 31, 2003. URL: <https://docs.broadcom.com/doc/PACKAGING-AN300-RDS.pdf> (visited on 6/12/2022) (Cited on page 53).
- [133] PCBWay. *Advanced PCB Manufacturing Cost Calculator*. 2022. URL: <https://www.pcbway.com/HighQualityOrderOnline.aspx> (visited on 6/12/2022) (Cited on page 53).
- [134] Infineon Technologies. *USB Controllers SDK*. 2022. URL: <https://www.infineon.com/cms/en/design-support/tools/sdk/usb-controllers-sdk/> (visited on 5/20/2022) (Cited on page 67).

Part II.

Appendix

A. Cypress CY7C68013A Voltage Level Translator and In-Circuit Programmer

A.1. Tweaking and flashing the firmware of an USBi programmer to convert the CY7C68013A

This chapter is dedicated to the setup and installation of the ICP based on the Cypress CY7C68013A MCU, like the one we used in the figure below.

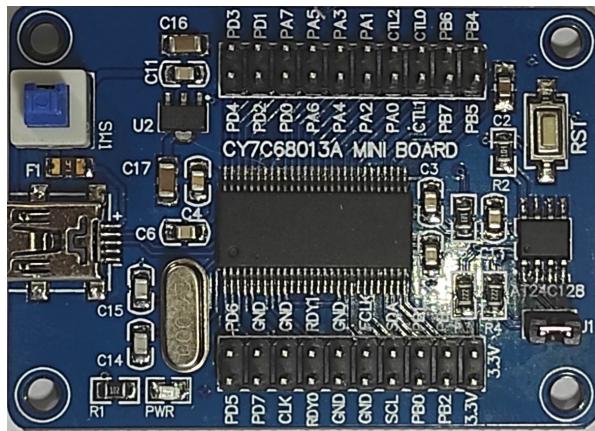


Figure A.1.: The CY7C68013A breakout board used in this project.

The setup required the following components to work:

- Step 1: Install the latest “SuiteUSB” USB controller SDK from Cypress¹ [134]. The SuiteUSB SDK includes two components which we will need: a generic USB device driver, and the USB control center for tweaking the parameters of the device.
 - Step 2: Plug the breakout board to the computer and open the Windows Device Manager. This can be achieved by pressing **Win**+**X** and choosing *Device Manager*. Another way to open the device manager is by opening the *Run* prompt when pressing **Win**+**R** and executing `devmgmt.msc` here.
 - Step 3: Inside the device manager, check under *Universal serial bus controllers* if a device has appeared called *Cypress FX2LP* or similar. It is unlikely that you will find it, but if it has appeared, you can skip the sub-steps below.
- 1) If the computer was unable to identify the device it is because it is missing the device drivers. Check under *Other devices* for any unknown devices.

¹Available at InfineonTechnologies website by registering an account with them.

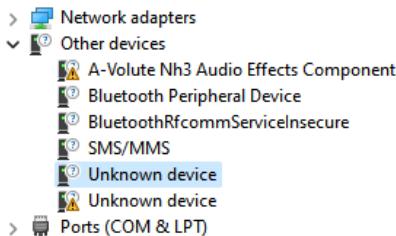


Figure A.2.: Multiple unknown devices in the windows device manager.

To make sure we have identified the appropriate device in this list, open **properties** inside the context menu by right clicking on the device, or double click on the device. Then check if the timestamp under **Details > Property > Last arrival date** coincides with time you have plugged the breakout board. Otherwise, check the remaining unknown devices if any.

If you still have not found the device, make sure the USB cable is plugged in correctly and that the device is powered².

- 2) Now that we have identified the device, in the same properties windows, copy the vendor and product IDs under **Details > Property > Device instance path**.

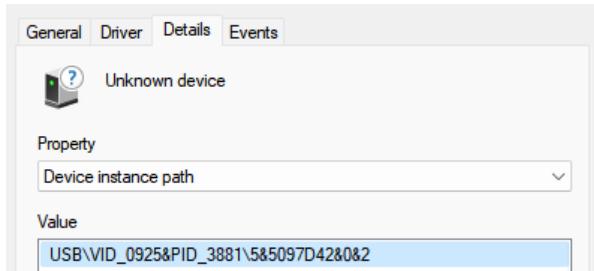


Figure A.3.: Vendor and product IDs of the breakout board.

- 3) We will now prepare the generic device driver from SuiteUSB so that our computer can identify the unknown device.

Navigate to `C:\Cypress\Cypress Suite USB 3.4.7\Driver\bin\wlh\x 64` or the directory you have installed SuiteUSB. Open `cyusb.inf` with a text editor and replace the strings `VID_XXXX` and `PID_XXXX` with the vendor ID and product ID from the previous step.

Also uncomment the four lines that begin with ";" by replacing that string with "%". These four lines are under the paragraphs beginning with `[Device]`, `[Device.NT]`, `[Device.NTx86]`, and `[Device.NTamd64]`.

- 4) Now we will attempt to install the modified drivers, but since we have tampered with them, a driver signature enforcement error will appear when attempting to install them, therefore we will need to disable temporarily the driver signature enforcement.

²Use the blue switch to power the device on and off.

One way of accessing the option to disable driver signature enforcement is through *Startup Settings*. A quick way to access these settings is by restarting the computer from the Start settings icon while holding $\lceil \wedge \rceil$ (shift) and clicking on the **Restart** button. After that, a black screen with the words *Please wait* should appear.

Another way of accessing the *Startup Settings* is through *Windows Settings*. Press **Win**+
I to open them, or search *Settings* in the Windows Search Bar, then go to **System** > **Recovery** >
Advanced startup and press on the button that says *Restart now*.

- 5) Once you have accessed the blue screen that says *Choose an option*, navigate to **Troubleshoot** > **Advanced Options** > **Startup Settings** > **Restart**. Wait until the computer restarts, then select the option *Disable driver signature enforcement*.

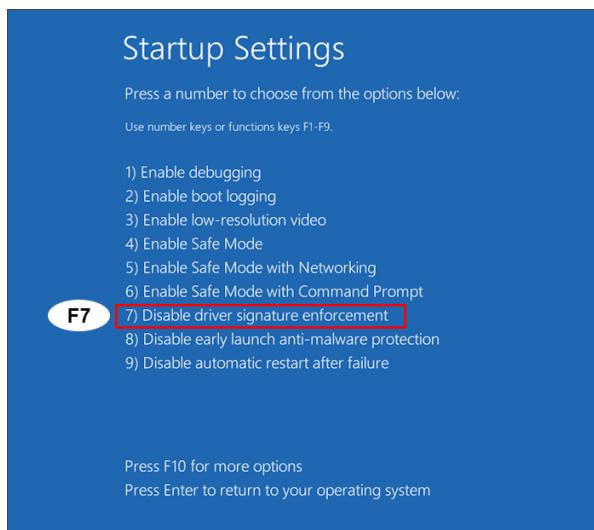


Figure A.4.: Windows startup settings showing to select the 7th option.

- 6) Once the computer has rebooted, reopen the Device Manager and proceed to install the driver. To do so, in the context menu of the device **Update Driver** > **Browse my computer for drivers** and navigate to the path where **cyusb.inf** was saved along the other auxiliary files.

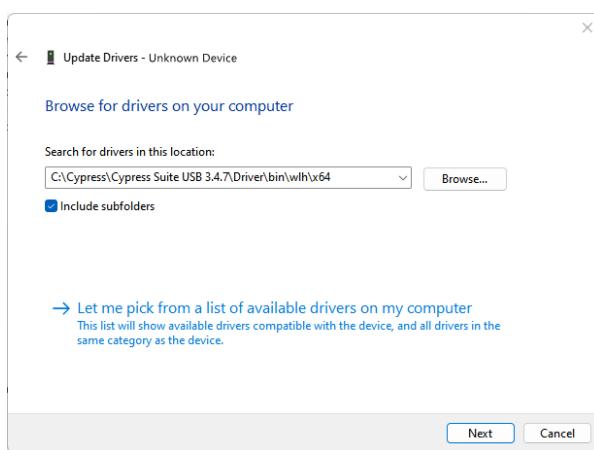


Figure A.5.: Window where the user is prompted to choose the driver.

A pop-up should appear warning the user that Windows was not able to verify the publisher of the driver. Install the driver anyway.

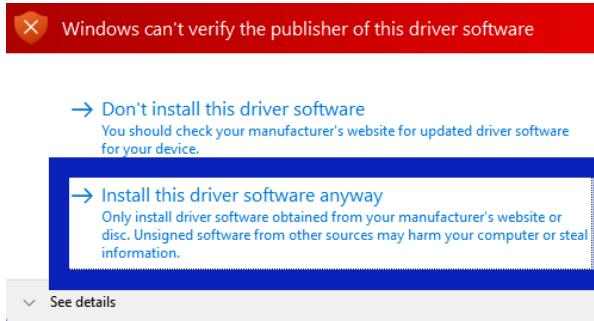


Figure A.6.: Warning that only shows when the driver signature enforcement is disabled.

- 7) Check if the device driver has been installed correctly like in the image below. If it has, we are ready to proceed to the next step.

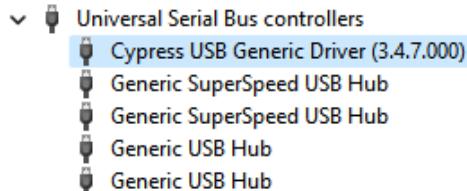


Figure A.7.: The device is now recognized in the device manager with the generic driver.

Step 4: Currently, the breakout board is just being recognized by the computer because we have installed the generic class driver. Open the “Cypress USB Console” by searching CyConsole in Windows Search.

In the toolbar select **Options > EZ-USB Interface**. Now press the button **Lg EEPROM** and browse for the firmware file [126].

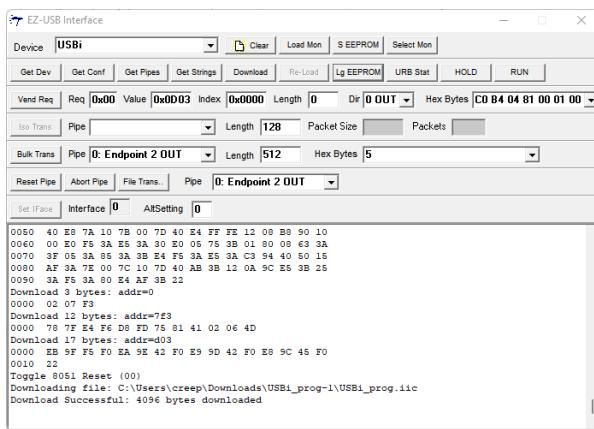


Figure A.8.: EZ-USB Interface after the EEPROM has been flashed.

Once the download is complete, return to the CyConsole and check that the Vendor ID and Product ID are the same like in the image below. This will allow us to install the firmware from SigmaStudio.

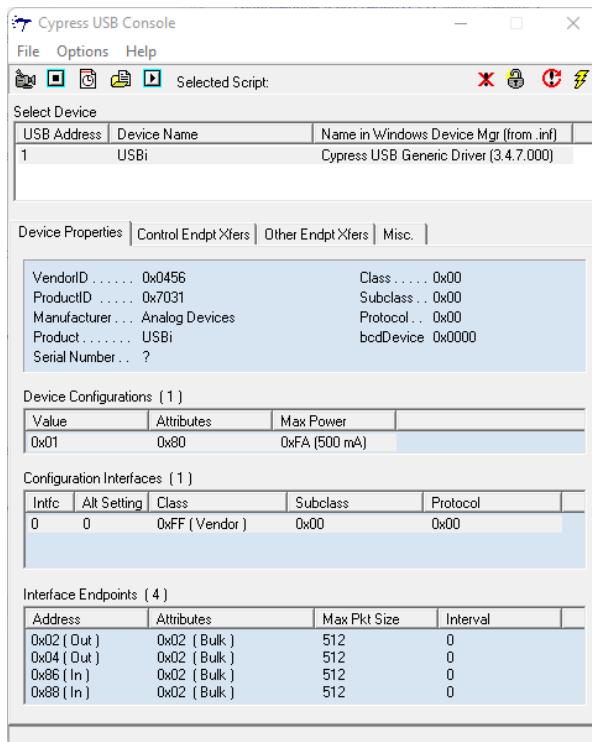


Figure A.9.: Cypress USB Console showing the properties of the USBi drivers.

Step 5: Reopen the Windows device manager and update the driver of the breakout board by replacing it with the driver from SigmaStudio. We can find the drivers in `C:\Program Files\Analog Devices\SigmaStudio 4.7\USB drivers\x64`. When the driver installation is completed, power off and on the breakout board so the OS shuffles the drivers.

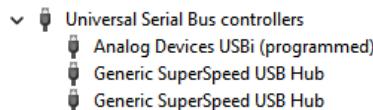


Figure A.10.: The device is now recognized in the device manager with the SigmaStudio driver.

Step 6: Now, open SigmaStudio, and open a new project with **Ctrl**+**N** or in the toolbar **File** > **New Project**. We will create a dummy project to test if the driver works correctly.

Place in the "Hardware Configuration" window from **Tree ToolBox** > **USBi** and check if the text is highlighted in green. Now place any of the ADAU processors from the **Tree Toolbox** and link it to the top pin of the USB Interface like in the image below.

If you three LED like in the image below, you should be able to see the I²C LED light up. If you change in SigmaStudio the address and protocol to SPI, and then press **F7** to **Link Compile Download** the project, the SPI LED should light up.

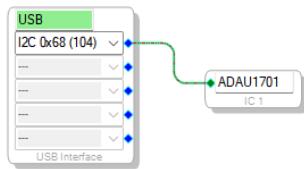


Figure A.11.: Demo in SigmaStudio showing the breakout board functioning.

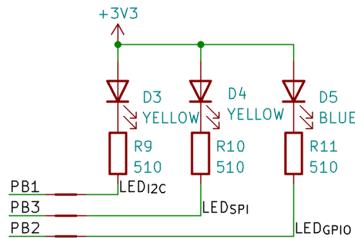


Figure A.12.: Schematic with connection to the status LED.

This is the end of this chapter. *Please remember to reboot the computer to reenable driver signature enforcement.*

A.2. 3D Object

Click on the image to open the view the 3D object. This requires a PDF reader with compatibility for version 1.7 or greater, like Adobe Reader.

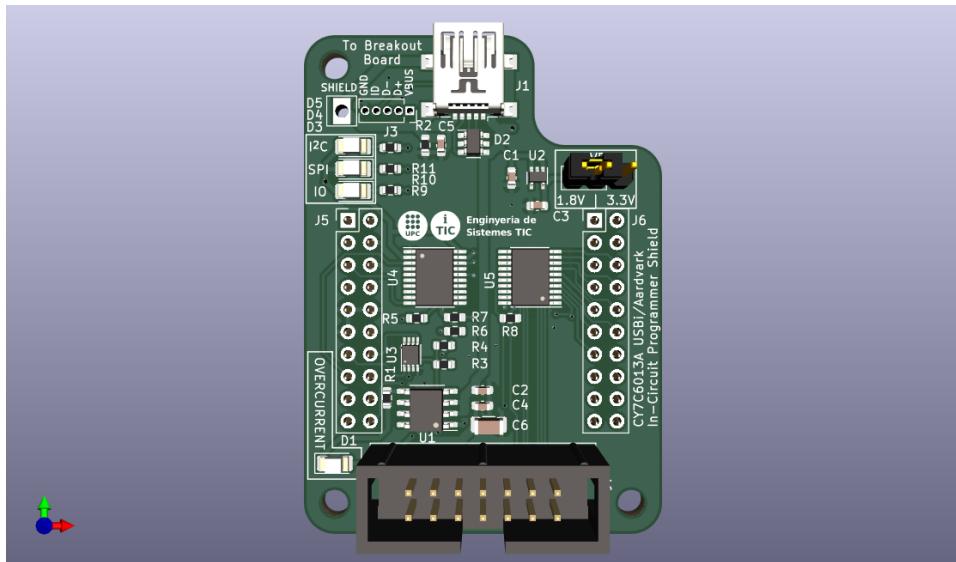


Figure A.13.: U3D object of the CY7C68013A Voltage Level Translator and ICP Programmer

A.3. The Schematic

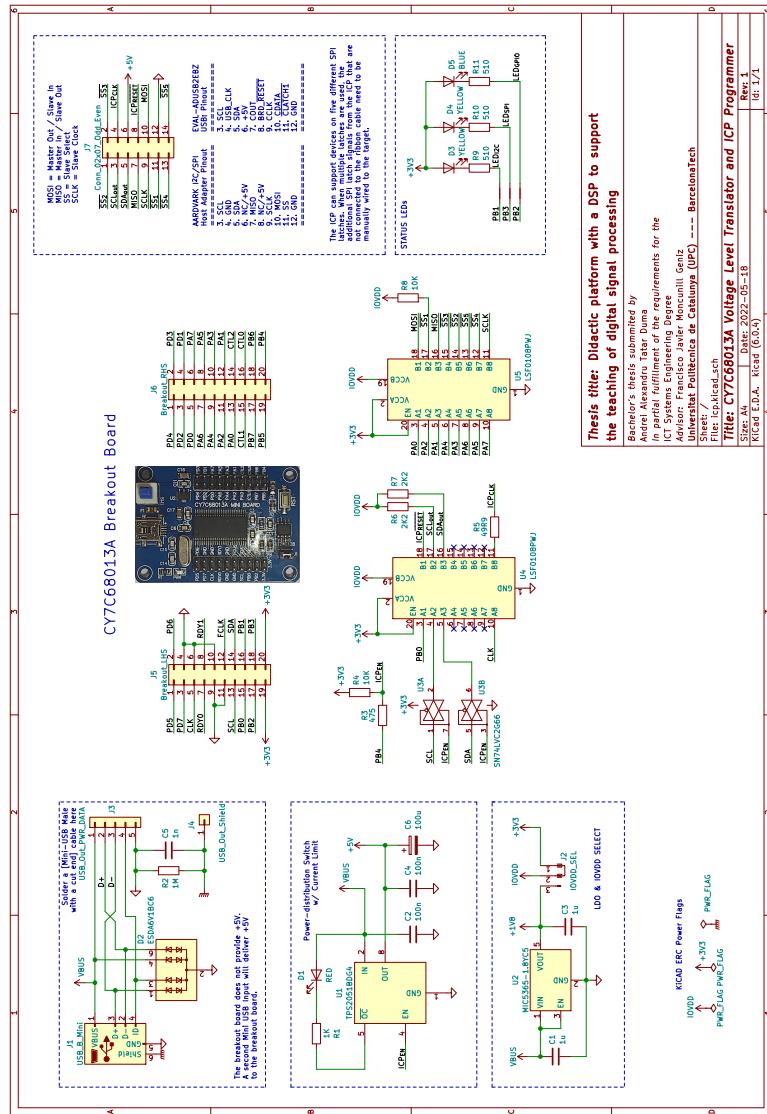


Figure A.14.: Schematic of the CY7C68013A Voltage Level Translator and ICP Programmer