Preface

Abstract

This document describes the development of an audio effects pedal which can be used in live applications. Both hard- and software will be covered. …

Table of Contents

Introduction

“summarize hard and software journey, flexibility goals etc, end application”

1 Hardware

After a first meeting with the company supervisor, Robert Manzke, it was pretty clear that a dual board construction was preferred. This method would allow for better reusability while maintaining a lower cost.

There is one 4-layer PCB which contains the microcontroller, memory, analog to digital and digital to analog converters, power switching and regulation circuitry and a display connector. Also provided but outside the scope of this thesis is a USB and SD card interface. This board, nicknamed "Cap'n MaGee", will from here on out be referenced as the compute board. This board is responsible for data processing.

The other board, which is a 2-layer design, features audio in- and output connectors, preamps and buffers, voltage regulation, battery management, Wi-Fi and the user I/O. This board, nicknamed "Aellö", will from here on out be referenced as the I/O board. This board is responsible for conditioning of analog input signals, battery management and makes the board mountable in a case.

The interface between the two boards is accomplished via a Mini PCI-e connector located on the I/O board. The compute board has fingers to match the connector. To fit the connector, the compute board needs to be 1mm thick and it is recommended to have a 45° chamfered border to extend longevity of the connector and matching PCB.

1.1 Compute board

1.1.1 Research

1.1.1.1 Microcontroller

As defined in the start-up report, the STM32H7 would be our microcontroller of choice, however this microcontroller exists in a couple of varieties, so research on features, necessary pin count and availability was done. Given all the peripherals and spare I/O, we came down to minimum I/O count of 125. This, together with solderability, led us to either the LQFP176 or LQFP208 package. Because of availability the STM32H743 in a LQFP176 was opted.

1.1.1.2 Memory

Prof. Manzke and I agreed on a minimum of 8 Mbyte SDRAM. Given the cost, 32 Mbyte was chosen. As for data bus width, 8 bit was preferred because of simpler routing.

1.1.1.3 CODEC

As mentioned before, the parts to convert analog signals into digital ones and digital into analog ones will be on the compute board. To lower the cost and space taken up on the PCB, an IC which contains both of those would be ideal. Several manufacturers make these, but because of their popularity, Cirrus Logic was recommended to me. In the end, their CS42448 stood out most, with its 6 inputs and 8 outputs, making for great reusability.

1.1.2 Supplies

The compute board can be powered off of 3 different sources. USB, Vaux and Vbat. USB power comes from the onboard USB connector. Vaux and Vbat go over the mini PCI-e connector. The Vaux input being a 5 V regulated input coming from an external connector on the I/O board. Vbat is provided by an internal battery, which has been boosted up to 5 V.

1.1.2.1 Power switching

When multiple power sources are provided, the board will automatically switch between them. The board has been designed so that Vaux gets priority, then USB and then Vbat. The hardware responsible for this is a power mux, more specifically the TPS2113A. When Vaux is applied, it shuts down the TPS2113A, letting its output float. Power to the system is then provided through the diode, D603.

When USB, in the schematic named Vbus, and Vbat are present, USB will be chosen as its power source. This is because the IC's Vsns pin is held high by Vbus, which in turn selects IN1 as the power source, connecting it to the OUT pin.

The 12 k pull-down ensures that Vsns will go low as the USB disconnects. When Vsns is low, our TPS2113A will choose whichever power source is higher in voltage. Given that this can only occur when Vbat is present, IN2 will be chosen and Vbat will be connected to the OUT pin.

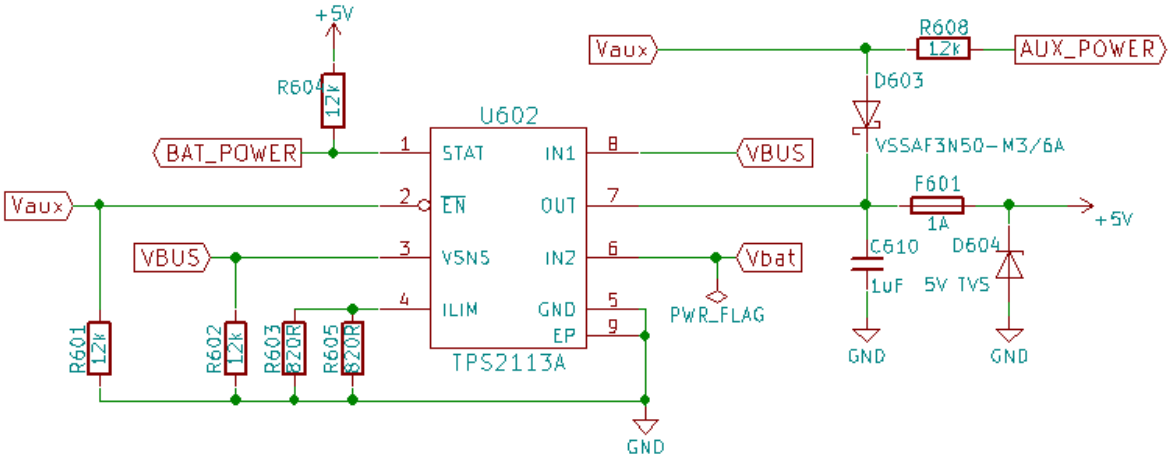


Figure 1: Power MUX schematic

1.1.2.2 Digital supply

Powering all digital components is the MIC23050. This buck converter steps down the 5 V input to the required 3.3 V. Its high switching frequency of 4 MHz makes it that the inductor and input and output capacitors can be low value, in term, taking up less board space.

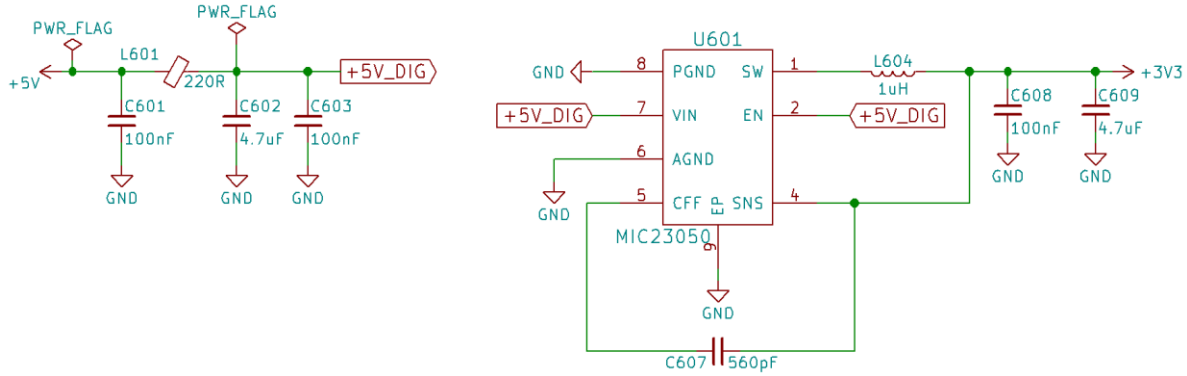


Figure 2: Digital supply schematic

1.1.2.3 Analog supply

The analog supply, providing power to the CODEC's ADCs[1]and DACs[2], is an ADP150. This LDO[3] was chosen because of its low output noise and high power supply rejection. As noise rejection goes down with increased frequency, an input filter composing of a ferrite bead and some capacitors is put in front of the input. This filter will give us another 48 dB of noise rejection at 1 MHz, increasing with frequency.

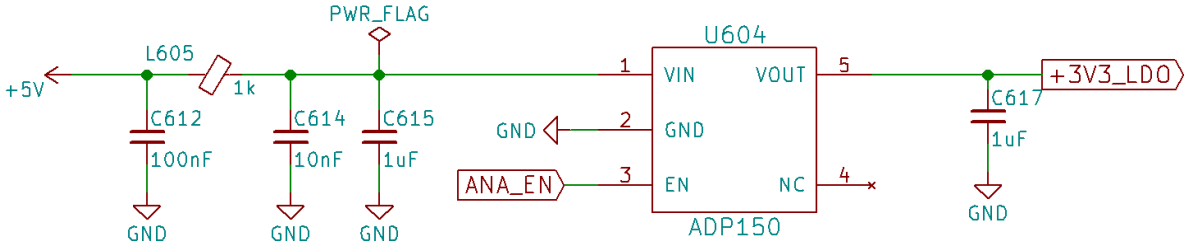


Figure 3: Analog supply schematic

[1] Analog to Digital Converters
[2] Digital to Analog Converters
[3] Low Dropout Regulator

1.1.2.4 Backlight supply

The screen intended to use with this board has a backlight which requires 19.2 V to reach its nominal 60mA. To reach this sort of voltage, a boost converter is used. As dimming the screen is a given feature these days, we went with the TPS61170. This converter has a reference which can be adjusted by programming, or using PWM[4] on the CTRL pin. In order to dim the screen linearly, the feedback pin gets the voltage from a low side current shunt, thus converting the IC in a constant current regulator.

However when a load with an impedance higher than 610 ohms would be connected across the TFT_LED+ and TFT_LED- terminals, the output would exceed its maximum voltage of 38V. To prevent this, a Zener has been fitted to keep the output voltage bellow 22V.

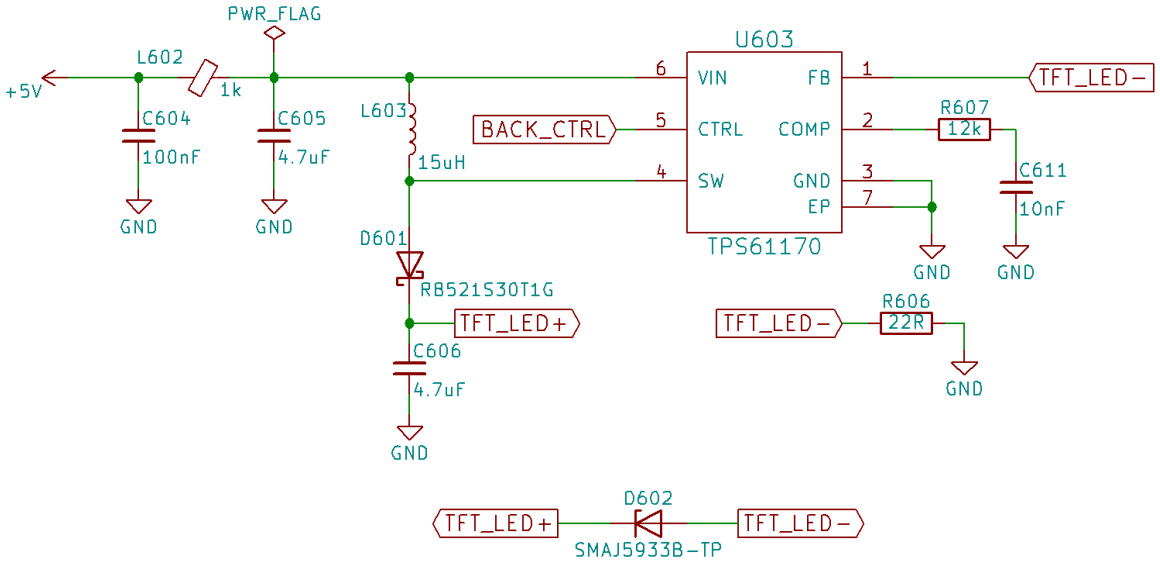


Figure 4: Backlight supply schematic

[4] Pulse Width Modulation

1.1.3 Microcontroller

The STM32H743II chosen has an 32-bit Arm® Cortex®-M7 core running at frequencies up to 400 MHz. It features a double-precision FPU[5], L1 cache, 2 Mbyte of flash and 1 Mbyte of RAM.

Peripherals used for this project are the LCD-TFT controller, Flexible Memory Controller, the Serial Audio Interface, SPI[6], I²C[7] and UART[8]. Other routed out interfaces include the USB High-Speed capable ULPI and SDMMC for SD cards.

The board also has 2 crystals to be used by the microcontroller. 1 high speed 16 MHz crystal and one low speed 32.768 kHz.

1.1.4 RAM

The ASC32M8SA-7TCN is the device giving us the 32 Mbyte of SDRAM. Those 32 Mbytes are divided into 4 banks of 8 Mbytes. The SDRAM is capable of speeds up to 143 MHz, however, this is limited to 100 MHz at the controller side by the FMC peripheral.

Communication between the FMC and the RAM is done over a parallel interface. This consists of CLK, 13 address lines, 8 data lines, chip, clock and write select enable and row and column strobe, as well as 2 pins for bank selection.

1.1.5 CODEC

The CS42448 is a 6 in, 8 out CODEC capable of sample rates up to 192 kHz in a 24 bit format. With its THD+N[9] of -98 dB in differential mode, its effective number of bits is 16 bit. This makes the board suitable for a wide range of applications.

The CODEC can be configured via an SPI or I²C interface. Using this, things like protocol, volume and input mode can be changed.

Samples are send over the Serial Audio Interface. This can be done using several protocols, however, TDM is recommended as this is the only one to gives access to all channels.

As the CODEC's analog supply is 3.3 V, the full scale input voltage lies between 1.75 and 1.95 V single-ended, and double that for differential inputs. Exceeding this will clip the digital output of the ADC. Harm to the IC will only occur when exceeding the absolute maximum ratings.

1.1.6 Screen

The board is designed to drive screens using the 24 bit RGB interface. More specifically the 5 inch NHD-5.0-8004800TF-ATXL#-CTP. Its resolution is 800 by 480 pixels and features a capacitive touch panel with multi-touch and gesture inputs.

Driving the display is done with the LCD-TFT peripheral. It's connected with 24 data lines, a clock input, line and frame synchronization signals and a data enable signal. They connect using a 40 pin FFC[10] connector with a 0.5 mm pitch.

The touch panel with its integrated controller communicates to the microcontroller making use of the I²C bus. A dedicated pin for interrupts is also present. This allows the touch panel to wake the microcontroller in the occurrence of an event. Connection of the touch panel and microcontroller are done over a separate connector. This connector is a 6 pin FFC with a 1 mm pitch.

[5] Floating Point Unit
[6] Serial Peripheral Interface
[7] Inter-Integrated Circuit
[8] Universal Asynchronous Receiver-Transmitter
[9] Total Harmonic Distortion + Noise
[10] Flexible Flat Cable

1.1.7 USB

The STM32H743 has 2 internal USB transceivers. These transceiver are only capable of Full-Speed USB. As the goal was to get Hi-Speed USB on this board, a Hi-Speed capable USB PHY[11] was added. The USB3300 from Microchip was chosen.

The PHY connects to the controller using the industry standard ULPI[12]. This 12 pin interface consists of 8 data pins, 3 control pins and a clock line.

The IC can be configured as a USB host, device or OTG[13] host. As this board has a USB B connector, OTG capability is lost.

1.1.8 SD Card

The SD card holder is directly connected to the microcontroller via the SDMMC peripheral. The peripheral can reach frequencies up to 100 MHz, allowing for transfer speeds of up to a theoretical 50 Mbyte/s.

The standard SD card interface consists of 4 data lines, a command line and a clock line.

1.1.9 EMI

1.1.9.1 Electromagnetic shielding

Because of the high frequency nature of all devices, a collective EMI gasket for all digital devices and switching power supplies was chosen. This helps with radiated emissions.

The Wurth Electronics' 36103505S SMD frame and its fitting 36003500S are made of tin plated steel. Although the manufacturer doesn't provide attenuation numbers with these 2 products, their pure tin variant do. Attenuation of 40 dB and higher can be expected between 100 and 400 MHz, with attenuation staying above 30 dB all the way until 3 GHz.

To reduce radiation from ICs to the bottom side of the PCB, a solid ground plane is fitted.

[11] Circuitry required to implement physical layer functions
[12] UTMI+ Low Pin Interface
[13] On The Go

1.1.9.2 Ringing

Ringing on a digital signal is the oscillation that happens after a sharp edge. This is caused by the parasitic inductance of a trace and the parasitic capacitance of the IC's pin. The parasitic capacitance causes a current spike whenever the signal changes state. This current spike together with the parasitic inductance generates a voltage spike, followed by oscillation.

Because of the parasitic capacitance, ringing gets worse as rise and fall time shortens. Increased ringing also causes increased electromagnetic radiation. As the compute board is full of high frequency components having short rise and fall times, resistors have been placed on several data lines to counteract ringing and thus radiation.

To reduce space and cost, small resistor networks were used. Each element having a resistance of 51 Ohms. This resistance combined with the parasitic capacitance of the trace and the pin of the IC form a low pass filter, attenuating high frequency components in the signal.

Other measures such as reducing parasitic inductance by keeping trace length to minimum are also taken. Increasing trace width would also have helped with this.

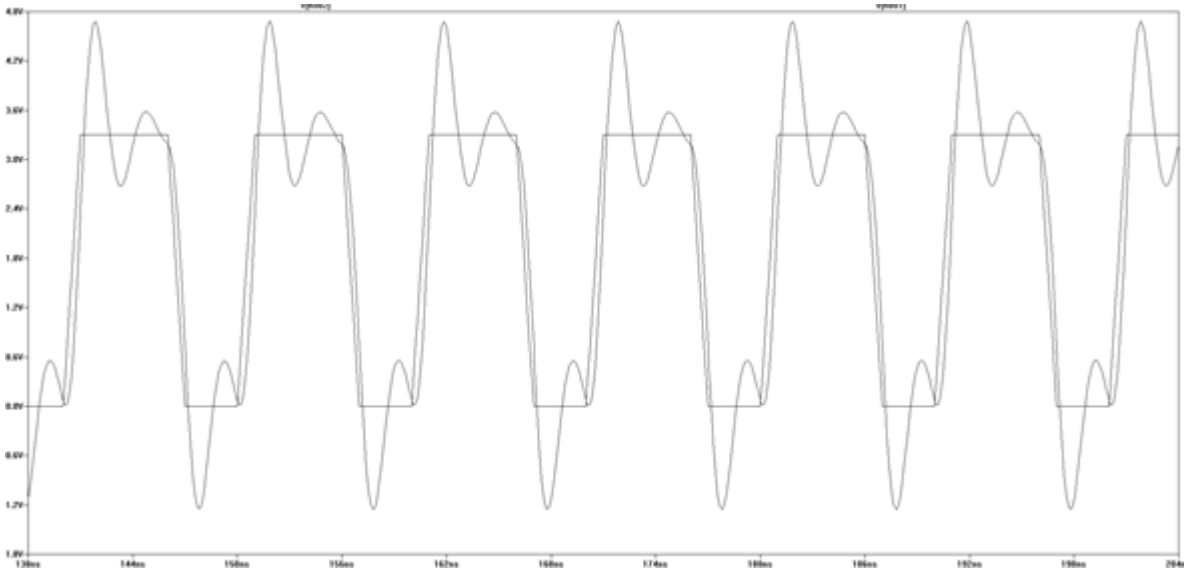


Figure 5: Ringing simulation

Often overlooked but also very important to reduce ringing is a low inductance power path to the IC. This can be accomplished by power and ground planes, wide but short traces and sufficient decoupling, all reducing loop area.

1.1.10 Signal integrity

Because, as previously stated, the board is packed with high frequency signals, it's good practice to match trace lengths to different ICs. This minimizes skew, ensuring that all data arrives at the IC at the same time and is read correctly. Maximum deviation from matching lies in the tolerances of both communicating devices.

As not to induce too much parasitic inductance, arced lines produces by a length matching tool have a diameter of at least 3 times the track width, preferably 5 times the track width.

1.1.11 ESD[14]

To protect the board from damage caused by static electricity coming from the outside, TVS arrays have been fitted near the SD card and display connectors. These will limit the voltage going in to the microcontroller. The USB is also ESD protected, implementing its protection in the USB PHY.

1.1.12 Test points

Various test points are present on the bottom side of the PCB. These, together with the Mini PCI-e connector can be used to program and test the device in a test bench if so desired.

1.1.13 Datasheet

As this device is meant to be reused in other projects, it seemed wise to construct a datasheet. The sheet consists of the board's features, pinout, test point and connector locations, physical dimensions and electrical characteristics.

This datasheet together with schematics are attached.

[14] Electrostatic Discharge

1.2 I/O board

Say something about connectors and inputs supported

1.2.1 Research

Although not super critical for the application, thought has been given to minimizing distortion and noise. This attempted by good PCB design and specific component choice.

1.2.1.1 Component placement

Notable on the board is that analog circuitry can be found in the top half of the board, while the DC/DC converters have been laid out on the bottom half. This can help separate the high frequency current spikes from digital circuitry from analog circuitry. More on this in the next topic.

1.2.1.2 Power and ground routing

It is generally desirable to have power and ground traces be low impedance, low inductance and high capacitance. Often this is done by a single plane across the whole board. Although not bad, this method could have some drawbacks when components aren't placed properly. Current spikes created by digital circuitries could create voltage drops, causing noise in analog circuits.

To prevent this from happening, a hybrid solution consisting of star power distribution and separate analog and digital ground planes in star form have been used.

1.2.1.3 Decoupling

Obviously, decoupling, the placement of capacitors near ICs, helps against noise, providing a low impedance energy source, in turn smoothing out current spikes seen by the rest of the system.

1.2.1.4 Capacitors

As there are quite a few different types of capacitors on the market, one might wonder what the difference is in audio applications and which one suits the purpose best. In order to answer this question, defining what makes a high quality system should be done first.

Generally, low noise, together with low distortion looks to be preferred, keeping the input signal clean. Low distortion seems to be related with high linearity.

As ceramics are rather nonlinear, changing in capacitance with frequency and applied voltage, these don't seem desirable in coupling applications. They are also prone to singing, mechanically resonating when a certain frequency is applied.

Common found in low cost audio applications are polarized electrolytic capacitors. Often provided with a DC bias, these capacitors provided distortion figures in the -80 to -90 dB range, making them suitable for many applications, including ours.

Stepping up a notch, bi-polar electrolytic capacitors provide excellent low distortion while keep cost and size low. This time, distortion ranging from -100 all the way till -120 dB, it is a good competitor against the next to be discussed film capacitors.

Film capacitors come in variety of materials. This part will be limited to PET and PP caps. The first one, PET or polyester, is the smaller one of the two. It tends to have slightly higher distortion than the PP or Polypropylene one, but at this point, distortion is so small that it doesn't matter for our application anymore. Film capacitors also tend to be bulky, making them impractical in some designs.

For this design, the bi-polar electrolytics are used as coupling caps, giving good value for money in a small package. Lower value film caps are used in the low-pass filter as at their capacitance rating, the price and size have come down quite a bit.

1.2.1.5 Resistors

Resistors can also introduce noise. Noise greatly varies on the resistive materials used. Carbon and thick-film resistors seem to be introducing more noise than thin-film, metal foil or wirewound resistors.

Thin-film resistors are a great low cost option. Their noise performance generally improves as temperature coefficient and tolerance lowers.

Wirewound resistors are low noise, but tend to be bulky and their inductance often isn't desirable.

Metal foil resistors stand out, providing lowest noise out of the bunch while having relatively small footprints. The downside of them is their cost.

Thin-film resistors are the better option for this application providing good balance between cost, noise and size.

1.2.1.6 Potentiometers

Single-turn potentiometers exist in a variety of materials. Most popular are the carbon composite ones. This is because of their low cost while still providing reasonable noise figures. However as they wear down, they are prone to turning noise.

Conductive plastic potentiometers counteract this problem, providing long life and low noise. Because of this, these are favored in higher-end audio equipment. These have been chosen as they are a good match for the preamps.

1.2.1.7 Op-amps

In the search for a suitable op-amp, multiple factors come into play. First of all, bandwidth must be high enough to support the gain needed. The gain multiplied by the maximum desired input frequency is the minimum gain-bandwidth needed.

For this board, it is also needed to support unity-gain stability.

Third, input noise is desired to be low, especially there where high gain can be present.

Lastly, having a THD+N figure in the datasheet is helpful to judge distortion produced by the amplifier. Not all datasheets show these numbers.

1.2.2 Supplies

1.2.2.1 Auxiliary power input

This barrel jack input is designed to be the main input of the device. It accepts the widespread 5.5x2.1mm plug often used for effects pedals and other applications such as the Arduino. The supplies for these aren't standardized, meaning that polarity can be swapped. To be able to work with both, a bridge rectifier is fitted. This input accepts input voltages ranging from 7 up to 18 V DC. AC power inputs aren't supported because input capacitance isn't big enough for the required load.

The TPS54332, chosen for its high input range and high switching frequency, is way overkill in terms of current, being able to deliver 3.5 A. At the time however, this was the cheapest well stocked buck converter available from Mouser. This IC steps the 7-18 V from input down to 5 V.

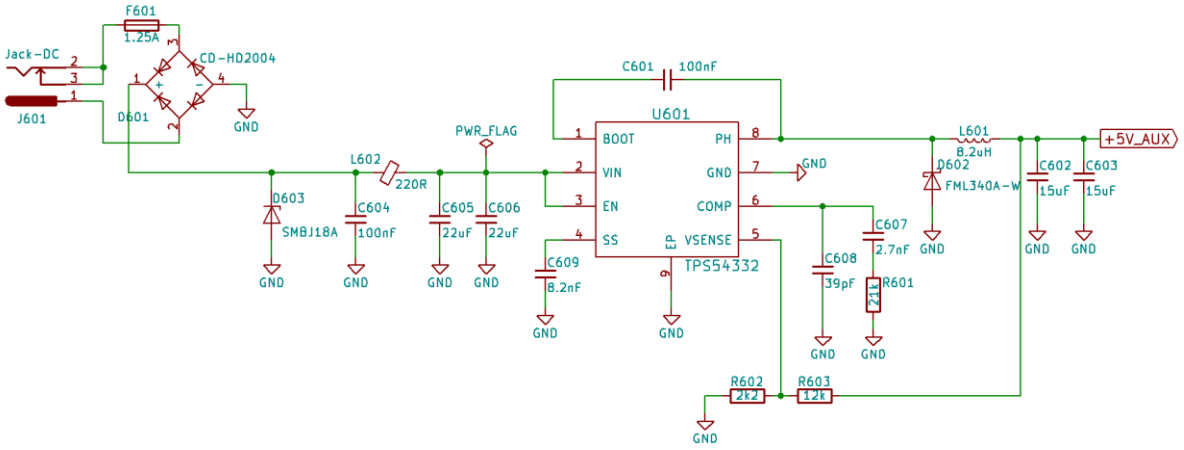


Figure 6: Auxilary power schematic

1.2.2.2 Battery input

This circuit protects the Li-Ion or Li-Po battery from undervoltage, overvoltage and overcurrent. It also protects the circuit against reverse voltage.

The AP9101C keeps a constant eye on the voltage across the battery terminals, comparing it to an internal threshold.

To protect against overcurrent, the voltage across the mosfets is measured. This voltage, present on VM, is then compared against an internal threshold. The maximum current depends on the on resistance of the mosfets.

If either of these thresholds are exceeded, the mosfets are shut down, preventing current from flowing.

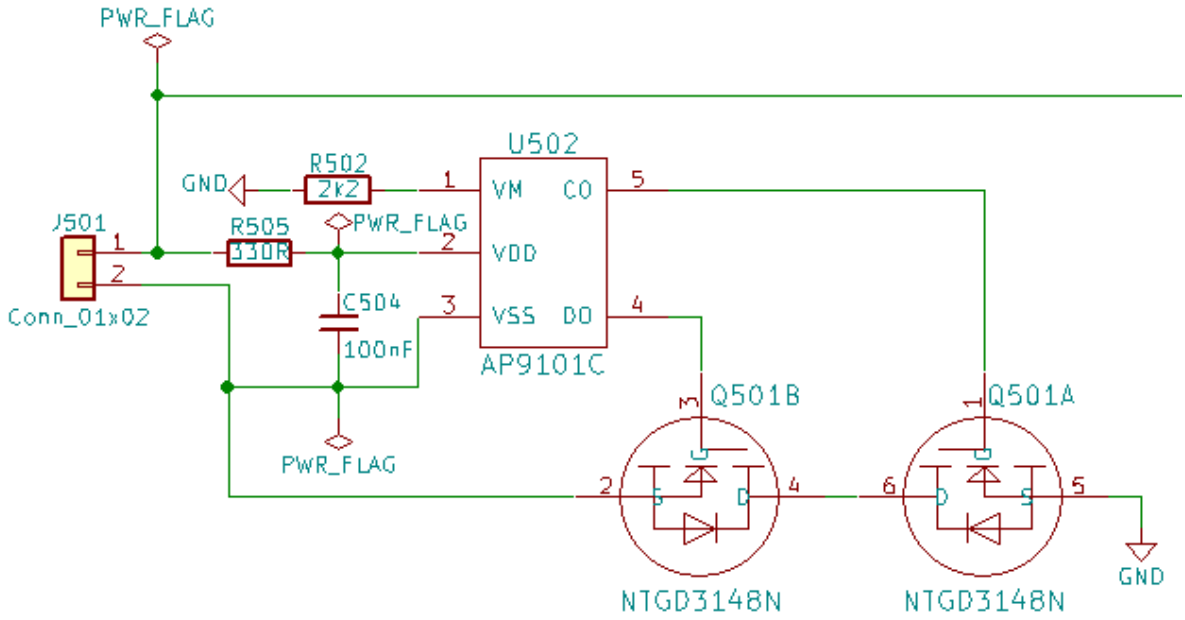


Figure 7: Battery input schematic

1.2.2.3 Battery charging

The DIO5158 has been chosen to support battery charging. It is configured to charge the battery at 812 mA from either the USB or Auxiliary power input. As the battery should last at least 2 hours, making for a minimum battery capacity of 2600mAh, temperature of the battery while charging isn't really a concern. The temperature protection is therefor just fed by a voltage divider.

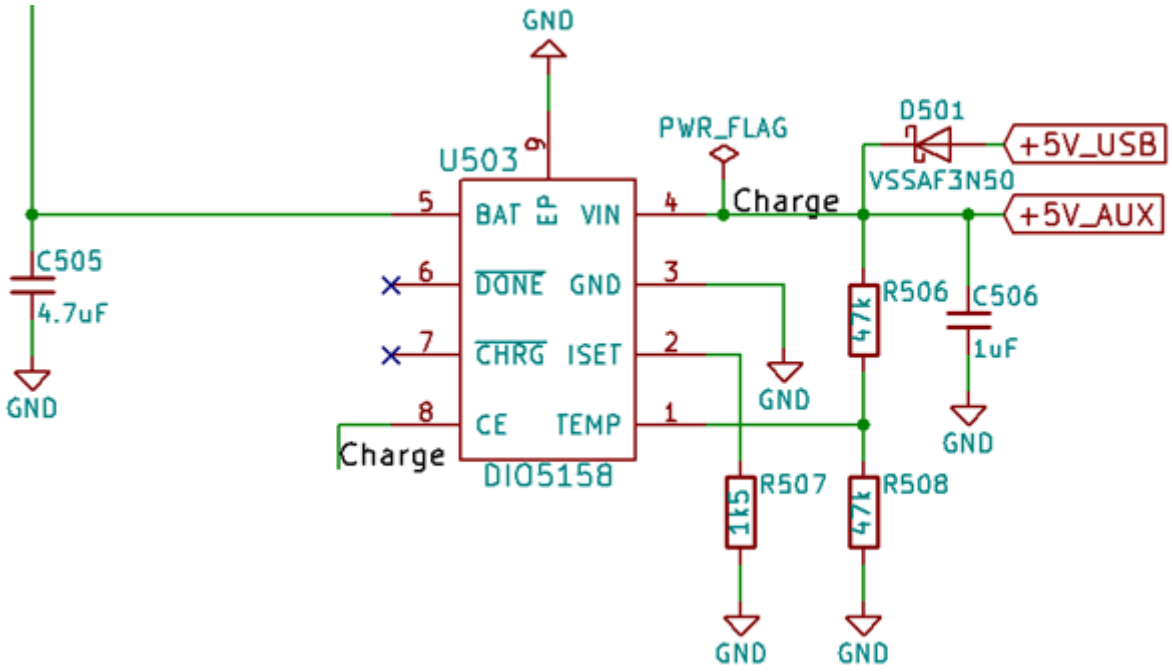


Figure 8: Battery charging schematic

1.2.2.4 Battery boost

The circuit shown below boosts the battery voltage up to 5V with a limited current of 1.5 A. The PAM2401's relatively low quiescent current and high efficiency decrease wasted battery capacity.

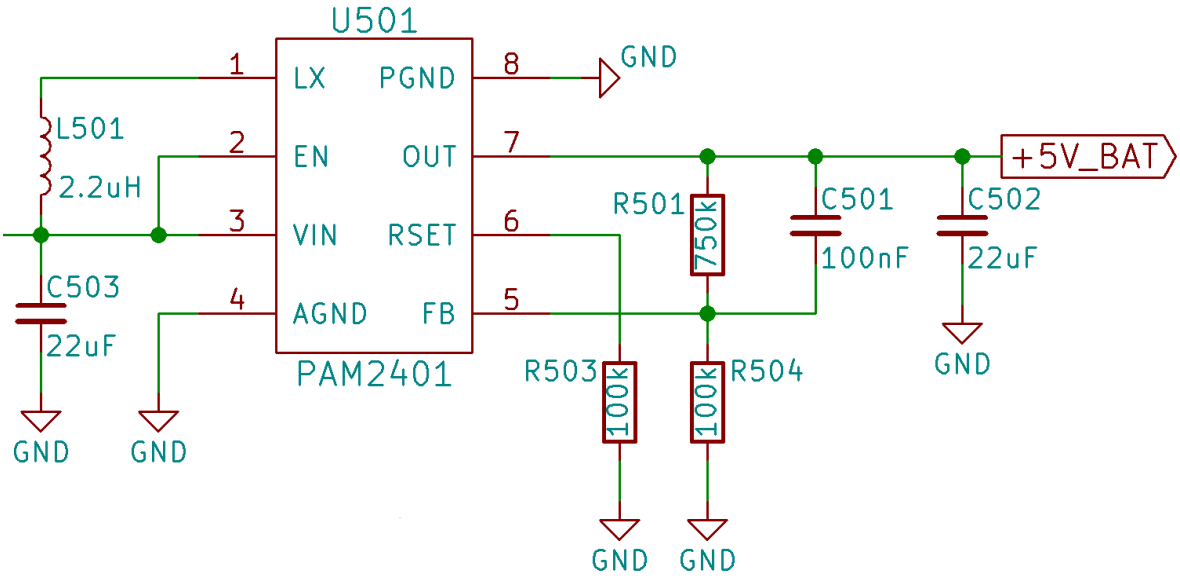


Figure 9: Battery boost schematic

1.2.2.5 Digital supply

This buck converter which is the same as the digital supply found on the compute board, is fed from the power switch output of the compute board. As with the digital supply on the compute board, it outputs 3.3 V and powers the ESP32 and is used for the switches and rotary encoders.

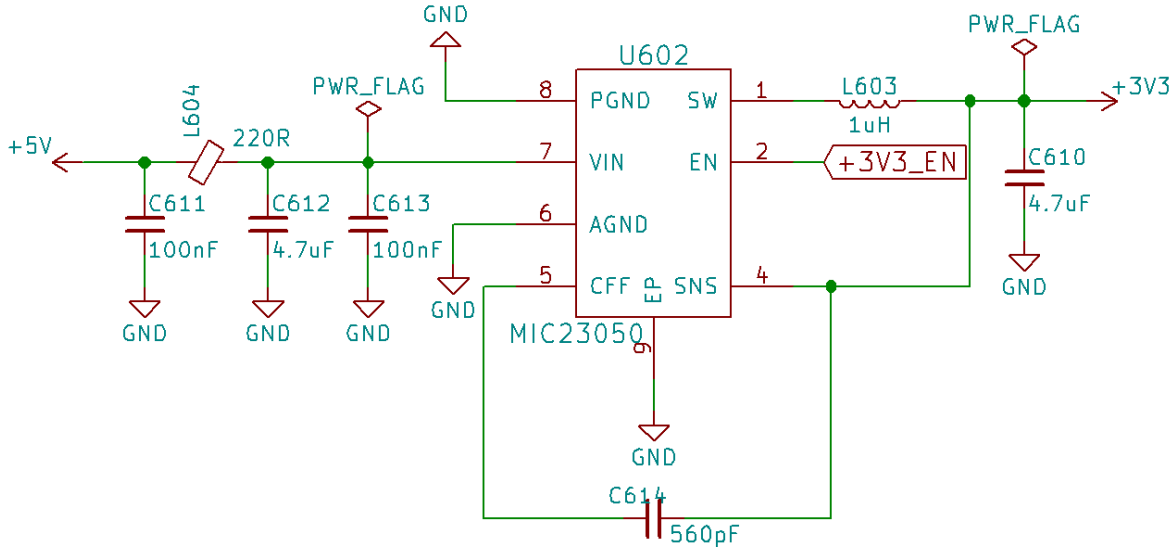


Figure 10: Digital supply schematics

1.2.2.6 Biasing

A voltage buffer provides the reference for inputs. The reference's voltage is the same as the center voltage of the CODEC. Although not completely necessary, the same has been done for the outputs. The supply for the op-amp is filtered through a 10 resistor and a 22 µF capacitor. This low-pass filter has a cut-off frequency of 4.5 kHz.

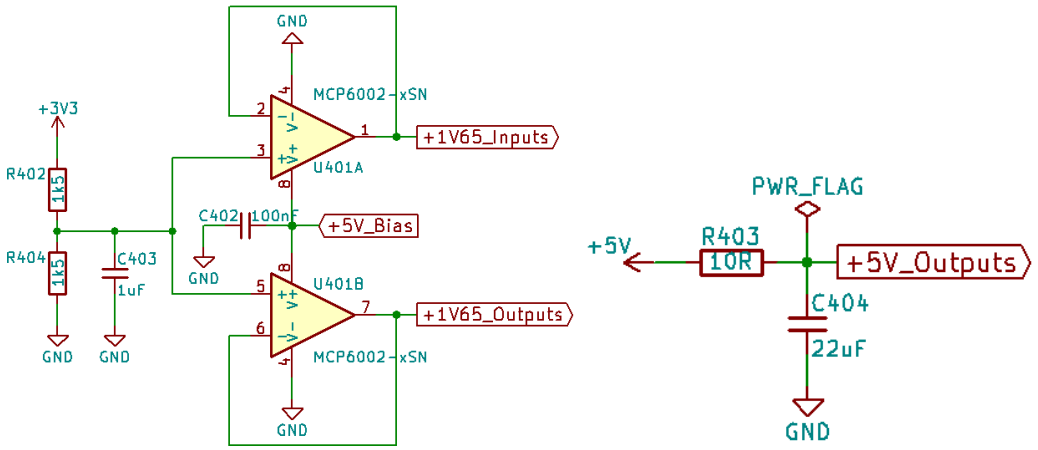


Figure 11: Biasing circuit

1.2.2.7 Phantom power

In order to keep cost low and reuse an IC, the TPS61170 boost converter was chosen again, this time generating 48 V for the phantom power used by condenser microphones.

Seeing that the TPS61170's maximum output voltage is only 38 V, a voltage doubler was added. D401 and C411 make up the first part of the boost converter. They generate a 24 V. This will provide an offset to the bottom end of C410. This point now contains the peak voltage from the inductor and the 24 V offset. Passing that through diode D403 and smoothing it out provides us with 48 V.

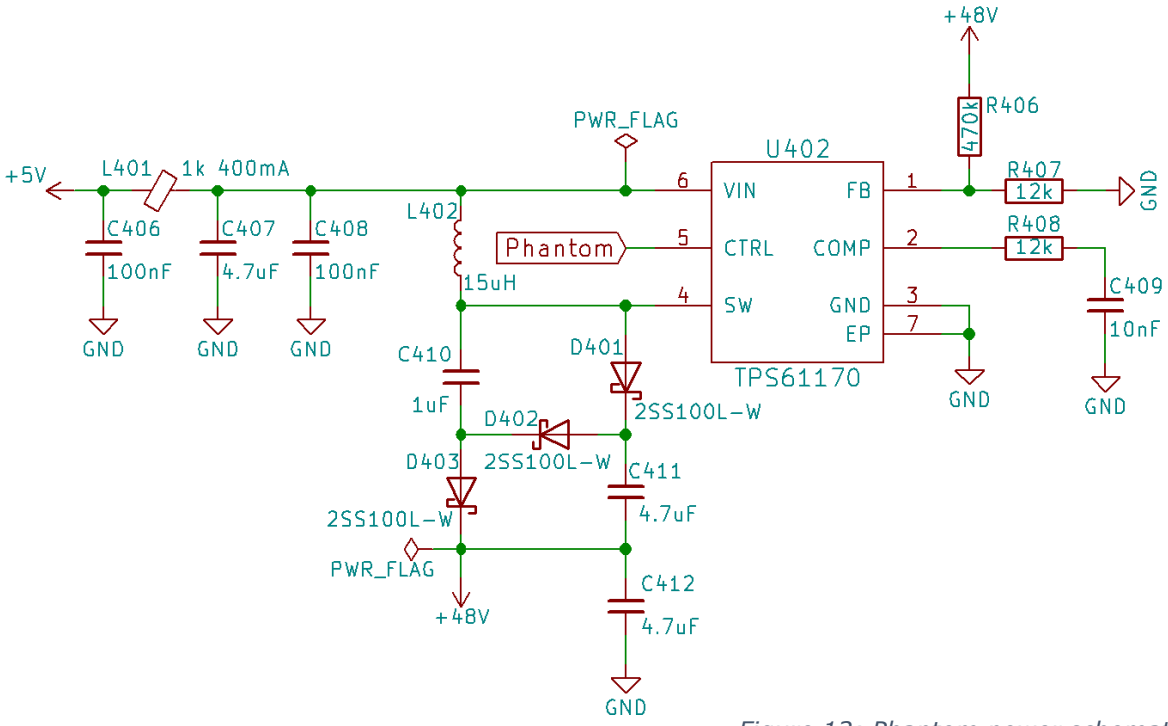


Figure 12: Phantom power schematic

Phantom power can be enabled for each input individually. This is realized using a complementary pair of transistors in a high-side switch configuration. The high value resistors used are chosen to limit the amount of current through the phantom power resistors in a short circuit event.

To reduce noise coming from the phantom power, the 6.81 k resistors used have to be matched within 0.1 % of each other. This will make it easier on the common mode rejection which reduces the noise from the phantom power.

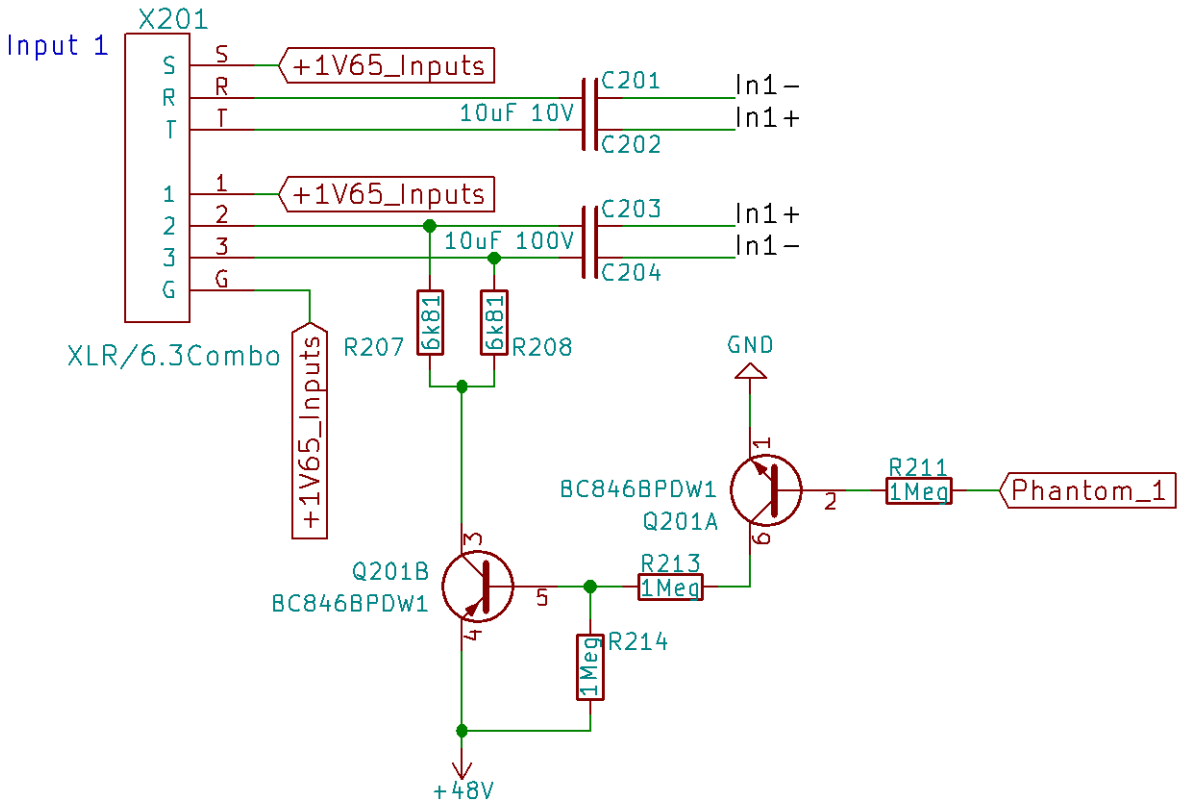


Figure 13: Input schematic

1.2.3 Pre-amp

Before the signal reaches the pre-amp, it is coupled through the aforementioned bi-polar electrolytics. Per input there is a pair of capacitors, the reason being that the 48V phantom power isn't allowed on the 6.3 mm jack part of the combo input connector.

After being coupled, a fixed 1 Meg impedance is introduced, together with a switchable 1.5 k impedance. The purpose of these to load the input source. The 1 Meg is a standard load for guitar, bass and various other instruments. 1.5 k is the normally used impedance to load down microphones.

The loaded signals then go through a 100k resistor, after which a switchable 47k is present. This forms a pad which, when turned on, attenuates the input by about -15 dB, depending on the source impedance.

As the goal is to support both guitar and a large variety of microphones, including dynamic ones, which can have outputs as little as 1 mV, a gain ranging from 1 V/V, otherwise known as unity gain, up to 1000 V/V is supported. With the audio bandwidth being up to 20 kHz, a minimum gain-bandwidth of 20 MHz is needed.

This considered, the MCP634 with its low noise and low distortion is chosen. Its common-mode and power supply rejection ratios are also sufficient for the application.

As seen in the figure below, the op-amp configuration used is that of an instrumentation amplifier. This configuration allows for a single-gang potentiometer to be used, minimizing gain mismatches. Its differential output also gives better THD+N at the CODEC side.

After going through a low-pass filter with a cut-off frequency of 23 kHz, it is kept between absolute maximum ratings for the CODEC using 2 back to back zeners. This can also be found right before the op-amp inputs, protecting them from unreasonable voltages.

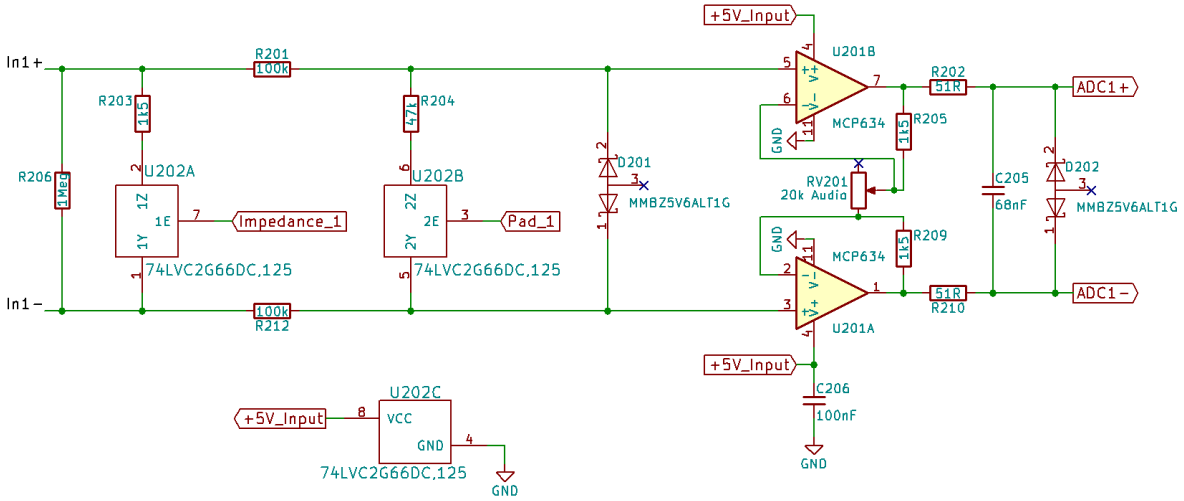


Figure 14: Pre-amp schematic

1.2.4 Auxiliary input

In order to play to music, a 3.5 mm stereo input jack is present. This can take signals from a phone or computer over an aux cord. The input is coupled and passed through a 23 kHz low pass filter. To protect the CODEC from overvoltage, a zener is fitted on each channel.

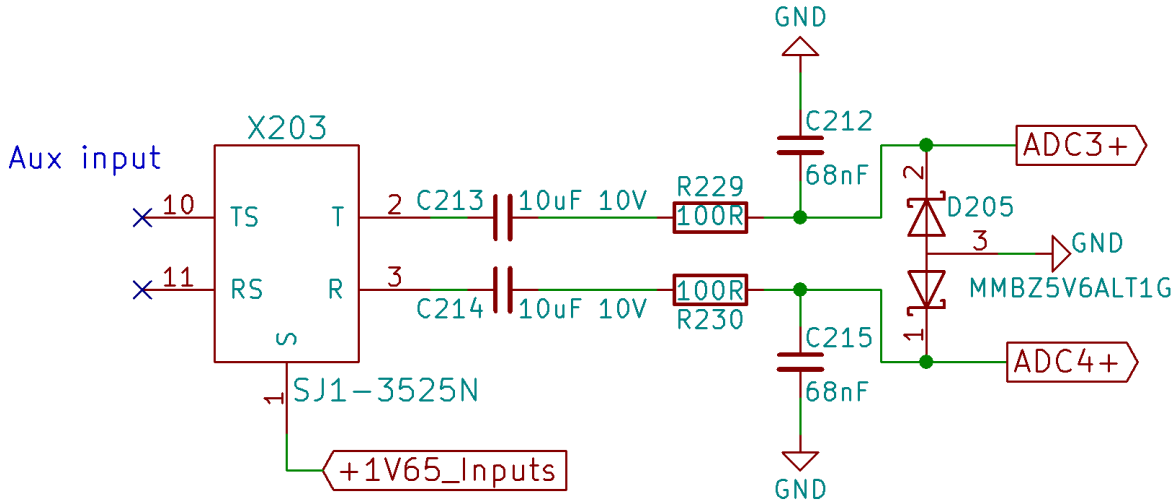


Figure 15: Auxiliary input schematic

1.2.5 Output

The main output is a 6.3 mm TRS jack. This sort of output supports both singled-ended and differential. The ring may be shorted to the sleeve.

Starting at the left of the schematic, the differential DAC output goes through a low-pass filter, reducing noise above 23 kHz. It is then passed through an op-amp configured as a voltage follower or buffer. The outputs are then protected by back to back zeners, protecting from outside voltage sources exceeding 3.3 V. The output is then pass through high voltage bi-polar coupling caps into the output. These capacitors allow for phantom power being put on the output. This could occur when an XLR to TRS cable is used.

The op-amp used here, the TLV9064 was chosen because of its known THD+N figure.

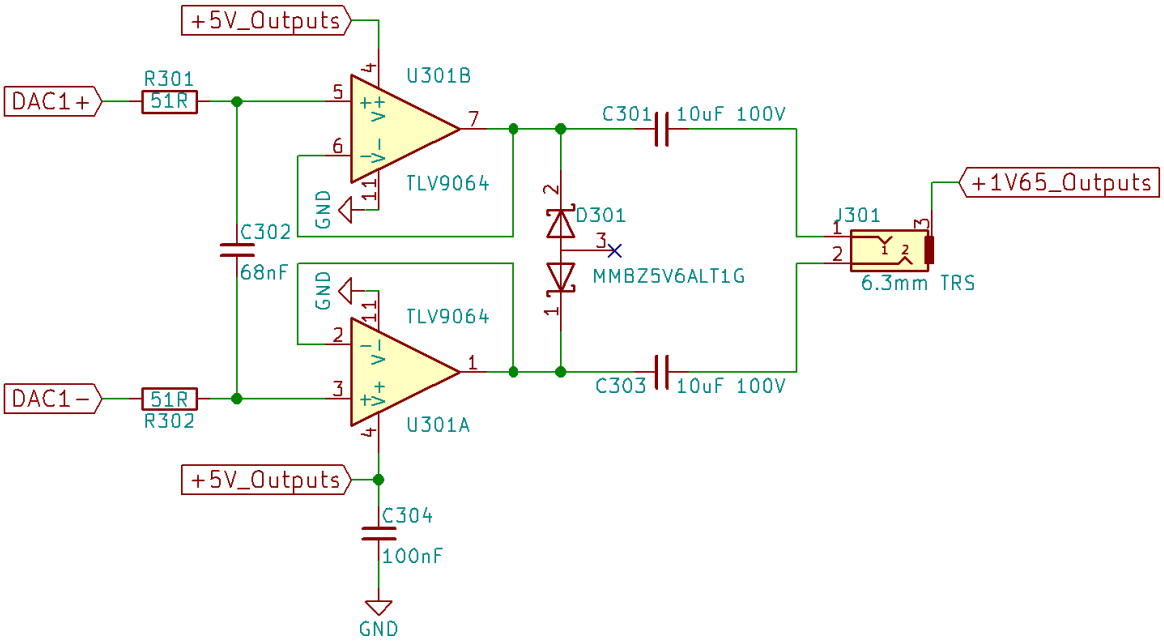


Figure 16: Output schematic

1.2.6 Headphone output

This output has 3.5 mm jack. It allows connecting headphones or external speakers. This circuit is almost identical to the main output one. It's different in that the outputs are single-ended, making for a stereo output. This time, the dual channel zeners are connected as one zener per channel. Smaller, less expensive output capacitors with a lower voltage rating have also been used as the chances of large DC voltages appearing on this output significantly smaller.

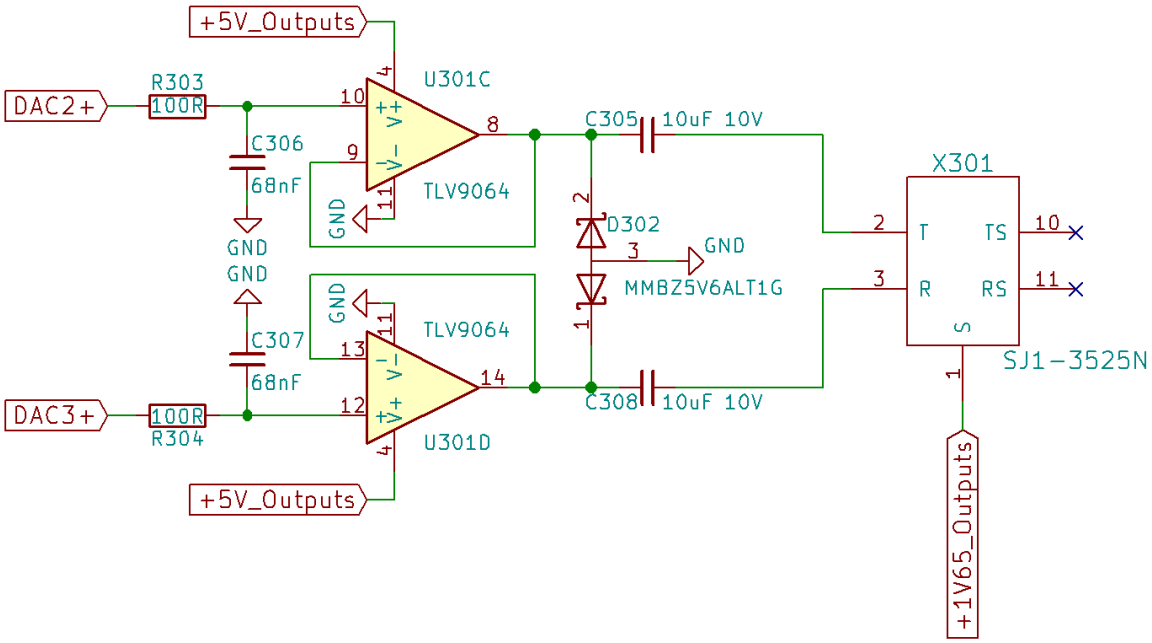


Figure 17: Headphone output schematic

1.2.7 Rotary encoders and switches

For rotary encoders, the Bourns PEL12D-4225S-S1024 was picked out. This rather expensive encoder features a bi-color LED and a switch. These LEDs are used as status LEDs of the different channels.

The E-Switch PS1024A is the switch used for toggling effects. This switch isn't as fancy as the rotary encoder.

All outputs are hardware debounced to ease software development.

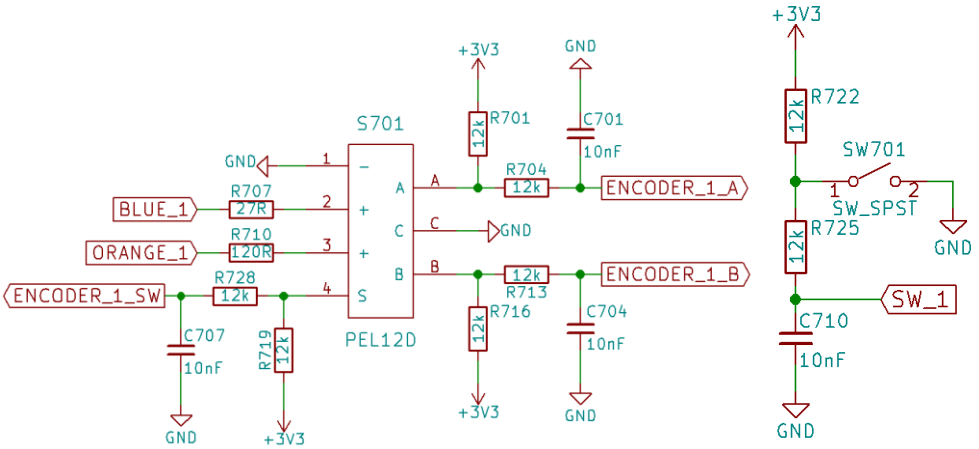


Figure 18: Rotary encoders and switches schematic

1.2.8 ESP32

To enable Wi-Fi on the board, either an ESP32 development board or the SMD ESP32S can be mounted. The ESP has been given the necessary buttons and header for programming.

The ESP32 controls the rotary encoders' LEDs and is also used to enable or disable phantom power and its individual channels.

Communication between the STM32H7 and the ESP is established over the UART interface.

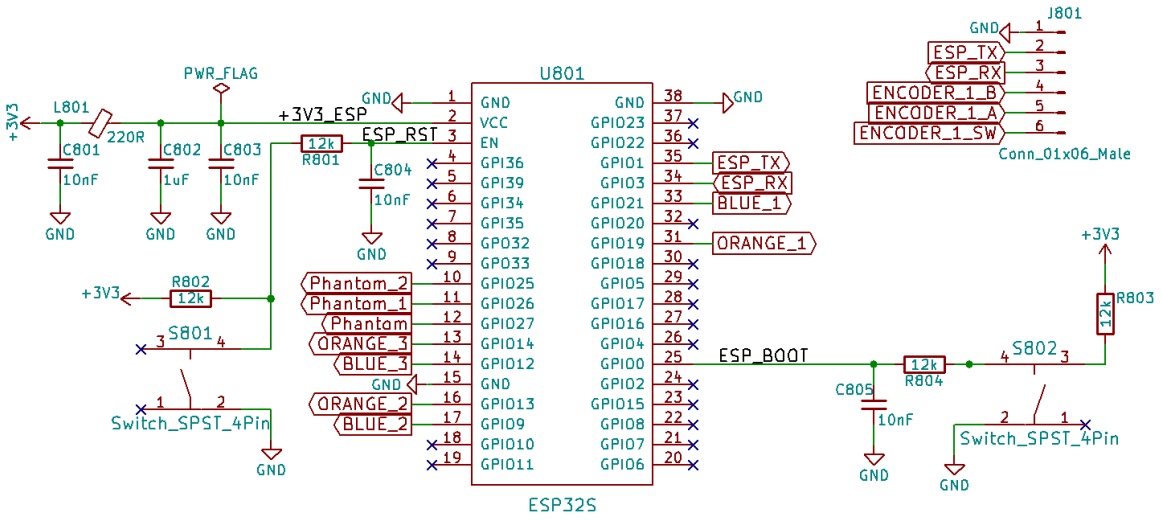


Figure 19: ESP32 schematic

2 Software

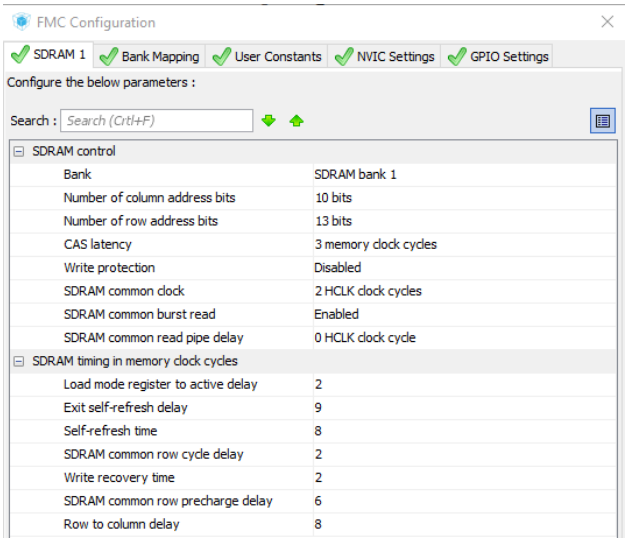
This segment goes over the programming of the STM32H7 and its peripherals.

2.1 Flexible Memory Controller

The FMC is a peripheral which can be configured to communicate with a variety of memory devices. SRAM[15], PSRAM[16], SDRAM[17] and NOR and NAND flash are all supported. On this board, the FMC is used to talk to the 32 Mbytes of SDRAM.

2.1.1 Configuration

Following figures show the configuration of the FMC according to the SDRAM timing specifications.



Symbol	A.C. Parameter		-6 Min.	-6 Max.	-7 Min.	-7 Max.	Unit	Note
t_{RC}	Row cycle time (same bank)		60	-	63	-	ns	
t_{RFC}	Refresh cycle time		60	-	63	-		
t_{RCD}	RAS# to CAS# delay (same bank)		18	-	21	-		
t_{RP}	Precharge to refresh/row activate command (same bank)		18	-	21	-		
t_{RRD}	Row activate to row activate delay (different banks)		12	-	14	-		
t_{MRD}	Mode register set cycle time		12	-	14	-		
t_{RAS}	Row activate to precharge time (same bank)		42	120K	42	120K		
t_{WR}	Write recovery time		12	-	14	-		
t_{CK}	Clock cycle time	CL* = 2	10	-	10	-		9
		CL* = 3	6	-	7	-		
t_{CH}	Clock high time		2	-	2.5	-		10
t_{CL}	Clock low time		2	-	2.5	-		10
t_{AC}	Access time from CLK (positive edge)	CL* = 2	-	6	-	6		10
		CL* = 3	-	5	-	5.4		
t_{OH}	Data output hold time		2.5	-	2.5	-		9
t_{LZ}	Data output low impedance		0	-	0	-		
t_{HZ}	Data output high impedance		-	5	-	5.4		8
t_{IS}	Data/Address/Control Input set-up time		1.5	-	1.5	-		10
t_{IH}	Data/Address/Control Input hold time		0.8	-	0.8	-		10
t_{PDE}	Power Down Exit set-up time		$t_{IS}+t_{CK}$	-	$t_{IS}+t_{CK}$	-		
t_{REFI}	Average Refresh interval time		-	7.8	-	7.8	µs	
t_{XSR}	Exit Self-Refresh to any Command		$t_{RC}+t_{IS}$	-	$t_{RC}+t_{IS}$	-	ns	

Figure 20: FMC configuration

The FMC is given a 200 MHz clock, resulting in a 100 MHz communication to the SDRAM. This makes the configuration a little slower as configuration was done with 7 ns clock time in mind. However, this shouldn't affect communication, instead making it more reliable, staying well within spec.

2.1.2 Initialization

Initialization steps for the SDRAM are adopted from the STM32F7 discovery's board support package. These steps configure the FMC's SDRAM command mode register and set the refresh rate in the timing register. These steps can also be found on page 800 of the STM32H7's reference manual.

2.1.3 Data transfer

After initialization, the SDRAM can be accessed using the HAL_SDRAM commands. The HAL_SDRAM_Read_8b and HAL_SDRAM_Write_8b can be used to transfer 8-bit arrays in blocking mode, meaning that it won't continue until the transfer is finished. Other data sizes can also be transferred using the _16b or _32b functions.

Data can also be transferred using the MDMA controller by calling HAL_SDRAM_Read_DMA or HAL_SDRAM_Write_DMA. This won't take as much resources from the CPU. The variable has to be located in the D1 or TCM section of the internal RAM for the MDMA to be able to access the variable.

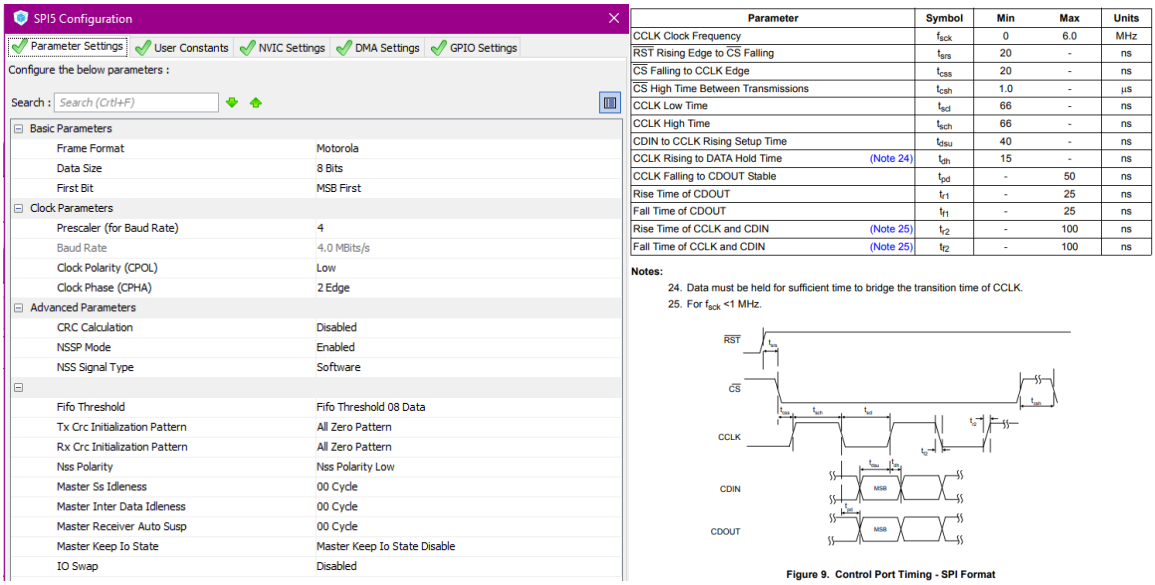
[15] Static Random-Access Memory
[16] Pseudo Static RAM
[17] Synchronous Dynamic RAM

2.2 SPI

The SPI bus is used to configure the CS42448 CODEC.

2.2.1 Configuration

Following figures show the configuration of the SPI5 peripheral according to the CODEC's timing specifications.



Parameter		Symbol	Min	Max	Units
CCLK Clock Frequency		f_{sck}	0	6.0	MHz
$\overline{RST}$ Rising Edge to $\overline{CS}$ Falling		t_{srs}	20	-	ns
$\overline{CS}$ Falling to CCLK Edge		t_{css}	20	-	ns
$\overline{CS}$ High Time Between Transmissions		t_{csh}	1.0	-	µs
CCLK Low Time		t_{scl}	66	-	ns
CCLK High Time		t_{sch}	66	-	ns
CDIN to CCLK Rising Setup Time		t_{dsu}	40	-	ns
CCLK Rising to DATA Hold Time	(Note 24)	t_{dh}	15	-	ns
CCLK Falling to CDOUT Stable		t_{pd}	-	50	ns
Rise Time of CDOUT		t_{r1}	-	25	ns
Fall Time of CDOUT		t_{f1}	-	25	ns
Rise Time of CCLK and CDIN	(Note 25)	t_{r2}	-	100	ns
Fall Time of CCLK and CDIN	(Note 25)	t_{f2}	-	100	ns

Notes:
24. Data must be held for sufficient time to bridge the transition time of CCLK.
25. For f_{sck} <1 MHz.

Figure 9. Control Port Timing - SPI Format

Figure 21: SPI configuration

As settings won't be changing constantly, this is just done in blocking mode.

2.2.2 CODEC initialization

After power-up, the fixed chip address is send followed by a logical 0, indicating that a write action is going to occur. Next, the memory address pointer is send. This contains the address to the register which is going to be written. It also has an bit which, when high, increments the memory address pointer. Doing this allows multiple settings to be adjusted without having to resend the chip and memory address.

In this initialization ADCs and DACs are configured, the protocol to the STM32H7's Serial Audio Interface is set up and transition settings for volume changes are specified.

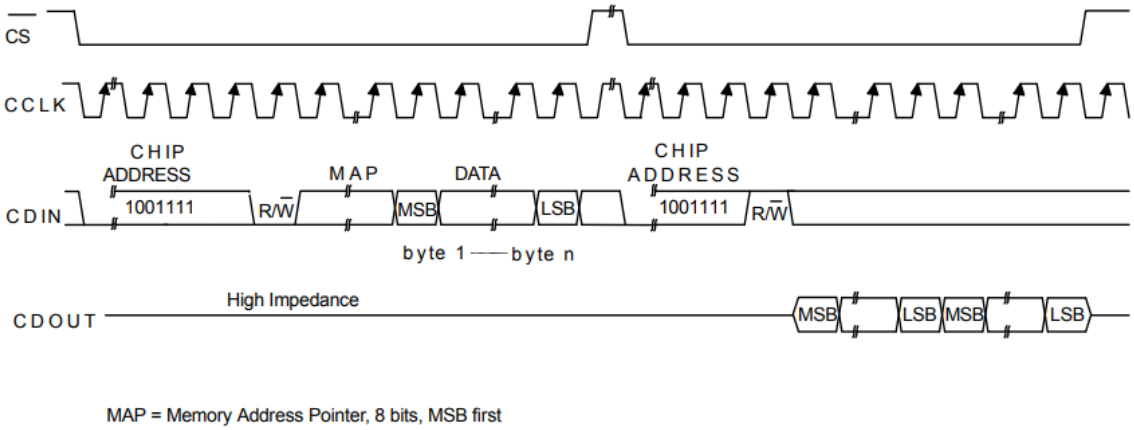


Figure 22: SPI communication

2.3 SAI

The Serial Audio Interface is a peripheral which can be configured to match many audio type protocols. Examples of such protocols are I2S, PCM, AC'97 and many more. The SAI is used to communicated with the CS42448 CODEC.

2.3.1 Configuration

The next figures show the SAI being configured in Time Division Multiplexing mode.

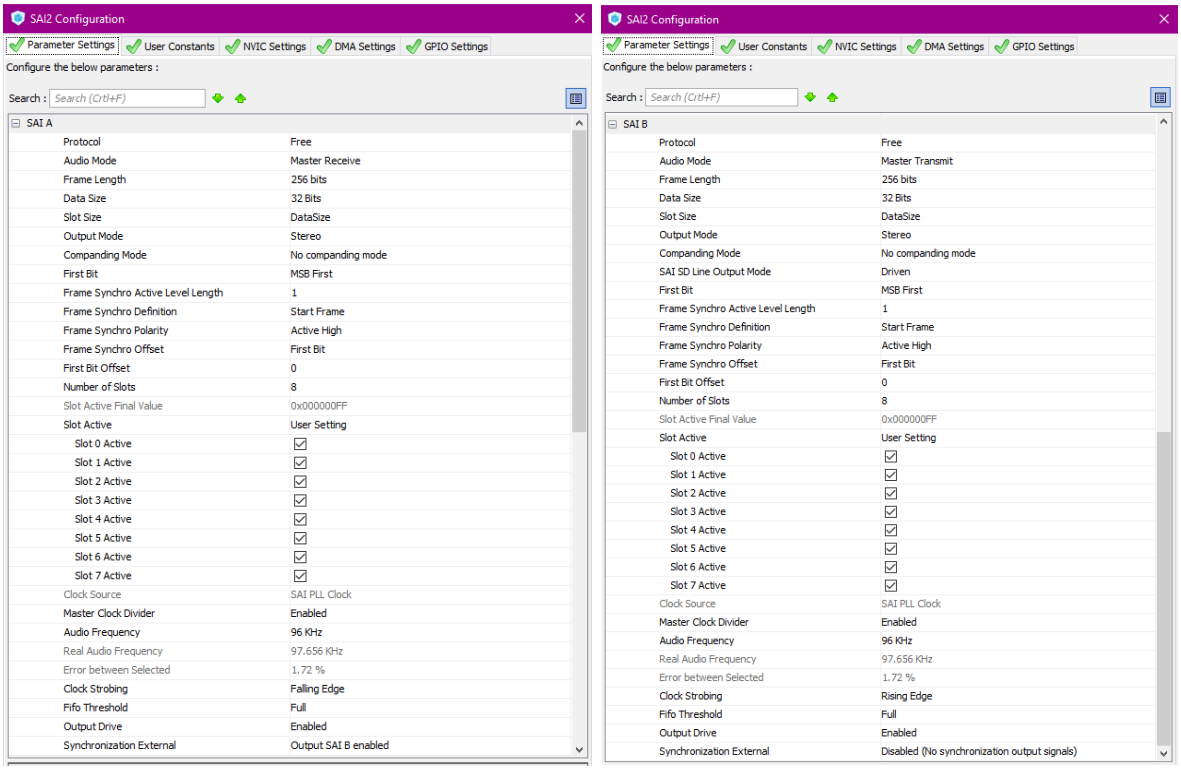


Figure 23: SAI configuration

This mode send all data over 2 channels. One for the ADCs and one for the DACs. Each channel has a frame select line, a clock line and a data in or output line. The frame select line, LRCK in the figure below, indicates the start for the next 8 data words. These data words, send over the data in or output lines, correspond to the analog in- and outputs.

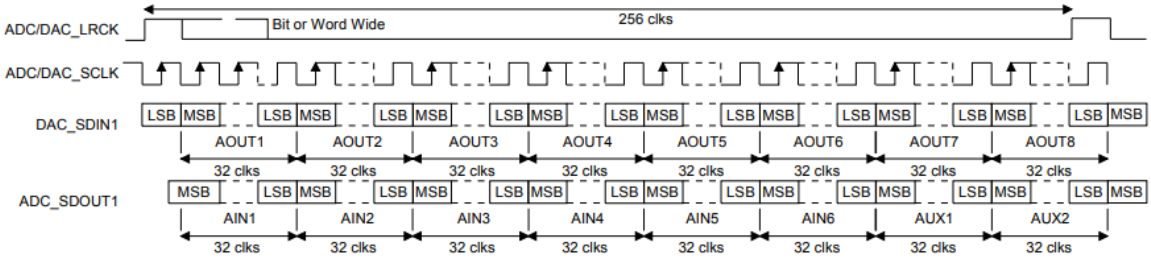


Figure 24: TDM format

2.3.2 Utilization

As the CODEC provides a constant data stream, it makes sense to use the SAI with DMA enabled. The Direct Memory Access peripheral copies data from the SAI's buffer to a variable without using the STM32H7's core. Because of segmented memory, the variable must lie in the same domain as the DMA's domain, D2. In this domain, 288 Kbyte of RAM is available.

Before using the DMA, it must be enabled and configured. It's set up for the DMA's data width matches that of the SAI. The circular mode allows for old data to be overwritten. The DMA's 4 word fifo can also be enabled to reduce traffic over the DMA. This is accomplished by buffering a maximum of 4 words before it is written to the memory.

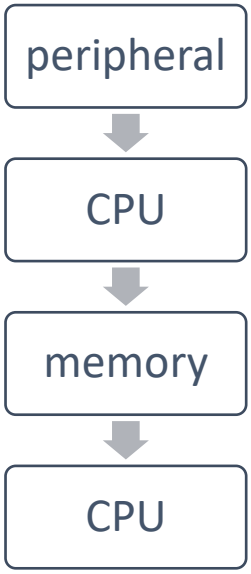


Figure 25: Without DMA

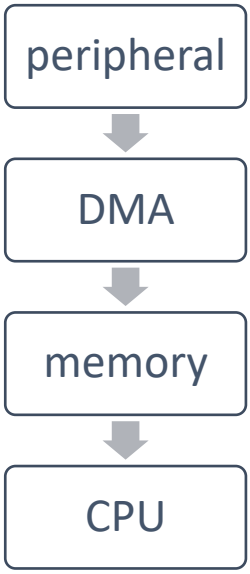


Figure 26: With DMA

The left figure shows the CPU having to show the transaction between the peripheral and the memory after which the CPU can use the data from memory.

The figure on the right illustrates using a peripheral using DMA. It shows the reduction of CPU usage, making more time for data processing.

To start the process, channel B of the SAI must be enabled. This enables the master clock output which is used by the CODEC as a clock sources for its converts and digital filters.

After this, HAL_SAI_Receive_DMA followed by HAL_SAI_Transmit_DMA can be called. These functions take the peripheral, data pointer and data size as parameters.

To increase performance, the data size should be more than just one data frame. Processing just one would make processing the digitalized audio very hard on the core. Using blocks of data gives more time for the processing. This however, does increase input lag. As a starting point, using a block of 24 samples, corresponding to 250 µs at the CODEC's sampling frequency of 96 kHz, is used. The 24 sample block makes the output lag half a millisecond behind the input, which is tolerable in this application.

This lag comes from the 250 µs sampling time plus the maximum 250 µs of editing time before the data will be send to the analog output. At the core frequency of 400 MHz, the editing time corresponds to 100 000 clock pulses.

As the block size is a variable, it can always be adjusted depending on the necessity of extra clock cycles needed to complete editing the waveform. Wanting to adjust this while running will make the structure of the program more complex. A solution where the user can adjust this depending on their preference could also be implemented.

2.4 Program structure