Imperial College
London

Imperial College London
Department of Electrical and Electronic Engineering

Final Year Project Report 2022

Project Title: The Design and Construction of an Optical Vinyl Disc Player

Student: Ruiqi Jiang

Course: EEE Meng Technical

Project Supervisor: Professor Andrew Holmes

Second Marker: Professor Stepan Lucyszyn

Abstract

The project aims to develop an optical vinyl disc player to play vinyl or shellac records contactless in real-time. The audio output is fully stereo. Since there is no contact with the record, the player does not cause wear to the records. The player is also easier to use, there is no need to adjust the balance, weights, etc. The player provides functions such as track skipping, raw, etc., that do not present on stylus players.

The OVDP targets vinyl/shellac disc collectors, archaeologists, and libraries. These users need to preserve the condition of their valuable collections. The OVDP provides them a way to enjoy their collections in good sound quality without devaluing them. The adaptive digital tracking system of OVDP can also play the cracked disc without gluing the disc.

The active groove tracking up to 100RPM also allows DJs to do scratching without scratching the DISC. The active groove tracking system does not generate any sound itself. All sound will come from the speakers. The active groove tracking system eliminates audio feedback issues even if the speaker output is loud.

Contents

Abstract 2

1. Acknowledgements 5

2. Introduction 6

3. Background 8

3.1. Existing products 8

3.2. Previous Project Work 8

3.2.1 The Read-Head 8

3.2.2. Audio Decoding Mechanisms 11

3.2.3. Focus Tracking System 13

4. Project Specification and Requirement 14

4.1. Overview 14

4.1.1. The Details of Real-Time Playing. 15

4.1.2. The Technical Spec of Sound Quality. 15

4.1.3. Groove Tracking and Maximum Trackable Center Deviation 17

5. Implementation 19

5.1. Module to be built and component choice 19

5.1.1. Choice of MCUs/FPGA 19

5.1.2. Choice of Motor driving signals 20

5.1.3. Choice of Other Components 20

5.1.4. Choice of software 20

5.2. Hardware Implementation 21

5.2.1. Top view module connection block diagram 21

5.2.2. Hardware connections at read head [21] 22

5.2.3. Hardware connection at Central processor [22] 23

5.2.4. Full circuitry and PCB layout 24

5.3 Software Implementation and Block Testing 25

5.3.1. Top view software architecture 25

5.3.2. Focus control system 26

5.3.3. Audio sample pick-up algorithm 33

5.3.4. Groove tracking control system 42

5.3.5. Audio Enhancer 44

5.3.5. “CD player” functions 53

5.3.6. Touch screen and groove visualizer 54
5.3.7. Play cracked mono shellac discs 55
6. Testing and Evaluation 57
6.1. Timing analysis, CPU utilization, and maximum sample rate 57
6.2. Glitch-less region 58
6.3. Headroom of Groove Tracking 58
6.4. Testing results on various discs 59
7. Future works 60
Reference 62

1. Acknowledgements

This project has been running in previous years. Many people have already made contributions to this project.

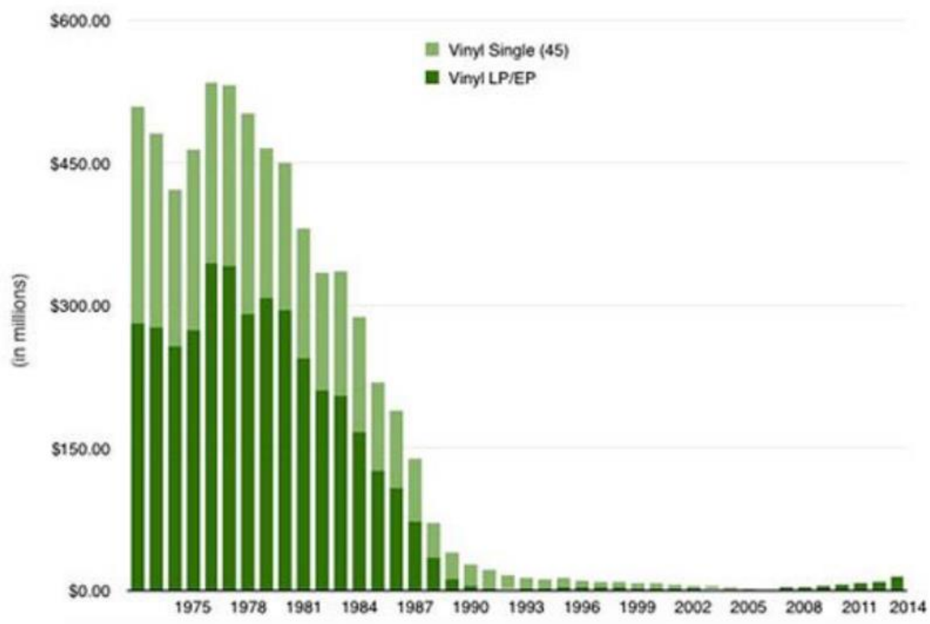
The new read head is designed and built by Professor Andrew S Holmes. He also provides support on adjusting the read head and other hardware throughout the projects. I want to say thanks for his tireless effort.

Mr. A. Welton has contributed to the development of the read head. Mr. T.J. Haworth built the old read head and the old audio decoding codes on FPGA. Mr. Mathew Wallace has designed photodiode amplifiers and comparators for focus tracking and audio signal pickup. He successfully decoded 21-second left channel audio from a real vinyl in 2008 and demonstrated the possibility of the player. I want to say thanks for their effort.

2. Introduction

The first Vinyl record was introduced by Columbia Record in 1948. It is a 12-inch disc with music signals stored inside the groove. It stores the music signals as an analog waveform. It rotates at a speed of 33.3RPM. It quickly replaced the classic 78RPM shellac discs. It became the most popular music source from 1950 to the mid-1970s. After the 1970s, digital music source such as CDs starts to take over Vinyl Discs.[2]

Figure 1.1 shows the market share of Vinyl discs from 1970 to 2014.



(Source: RIAA, acquired from Resnikoff, 2015)

Figure 1.1 market share of Vinyl discs from 1970 to 2014.

The market share reaches a peak at around 1976 and then decreases. This means that the original records from the 1950s to 1970s are mostly stored in Vinyl. To listen to these original records, a vinyl player is required to play directly, or otherwise, a vinyl player is required to convert them to digital files.

The market share of Vinyl starts to increase again after 2007 and the rate of increment gets bigger afterward. [1] This implies that even though digital audios and streaming are available, there is still a growing demand for vinyl discs and vinyl players.

To satisfy the vinyl player market, one must understand why specific people started to prefer vinyl again. There are mainly five reasons [3].

Reasons for buying vinyl	Correspond vinyl player feature required
Warm and a "feeling of real" sound of vinyl, no digital compression (in mp3, aac, etc)	Less quantization and digital processing in the signal path. Play the music in real-time. Preserve the warm and analog feeling of an old vinyl player.
Audiophile's hobby. Higher sound quality with expensive vinyl players and records	Preserve fidelity, less THD, flat and wide frequency, and phase response, achieve high audio quality through using the expensive components. No sound quality degradation after each playing.
Hold and own a vinyl that one has dreamed of for a long time	No damage to the vinyl after playing. The vinyl disc player should not be too expensive.
Vinyl experience, listen to music on a vinyl player while enjoying the evening.	No damage to the vinyl after playing. The vinyl disc player should not be too expensive. Play the viny in real-time.
Value of Vinyl	No damage to the vinyl after playing.

Table 1.1 Identify product direction through the market.

In summary, those who bought vinyl, except audiophile which took up only a small portion of the markets, would prefer a vinyl machine that:

1. Does not wear the vinyl when playing.
2. Must be able to play the record in real-time.
3. Being able to preserve the feeling of listening to vinyl, such as the visual feeling and the warm sound signature of the vinyl. The sound signature doesn't have to be HIFI.
4. Not too expensive for most user

Traditionally, vinyl is played using a turntable with a 'stylus' (needle) which rests in the record groove as it rotates. The variations in the groove shape cause the stylus to oscillate, and this movement is picked up by tiny coils which generate an electric current. This is then amplified to reproduce the audio. The main disadvantage of this method is that the stylus causes wear so that the record degrades on each playing. [4] This is not preferred by most users in the market.

As a result, a low-cost optical vinyl disc player should be developed. Such devices should read the music waveform from the groove without contacting or damaging the vinyl disc.

Due to this non-contacting feature, such devices can be used in libraries and museums. High-valued and non-replicable vinyl can then be played to the audience without damaging the original disc. Vinyl discs with broken and damaged tracks can also be played on such devices by utilizing modem digital signal processing techniques.

When cost is not an issue, such devices can be upgraded in the future with expensive components and higher-speed processors to meet audiophiles' needs.

3. Background

3.1. Existing products

Having the four basic features of a market-preferred contactless vinyl player, it is time to look at existing products and justify their pros and cons.

The only available commercial laser vinyl player is from ELP. It costs $15000. The cost of this product may only be accepted by audiophiles who make up only a small portion of the market.

The technology used in the ELP laser vinyl player is not published.

The high production cost of ELP laser vinyl player could be due to the following facts. The ELP uses laser beams to track and play audio contents inside the groove. The ELP laser vinyl player has a pure analog audio path [5] which means that it must maintain two lasers pointing exactly on two sides of the groove and amplify the signal they picked up directly. There must be other sets of lasers that precisely control focus and track the two edges of the groove. At least five lasers with very precise circuitry are required in the ELP laser turntable. These result in high sound quality but also high price.

Using only a laser and a scanning mirror to do groove tracking and audio playback could cut down the cost. A laser scanner scans the surface of the vinyl to digital time or image signal, then, tracking and audio playback are processed through these scanned images. This method required less tracking accuracy and tracking speed than the ELP's approach. This reduces the cost. Since the audio is coming from a digital image, the signal path is no longer pure analog. For most consumers except audiophiles, this will not be a problem, given that the basic fidelity is preserved through linearity and frequency response correction.

3.2. Previous Project Work

Mr. A. Welton [6] demonstrated that an optical approach using an oscillating micro-mirror and a laser was feasible in 2004. Mr. T.J. Haworth [7] continued the project, constructing the laser read-head (details in 2.2.1) and writing code for an Altera Cyclone-II FPGA (Field Programmable Gate Array) development board in 2005. In 2007, Mr. Matthew Wallace [4] developed a focus tracking system that can maintain the distance between the read head and vinyl at 7mm with an error <0.1mm (2.2.3). He also builds circuits board for amplifying signals from the photodiode, circuits for driving voice coil motors, and Schmitt trigger circuits for groove detection.

By the end of Mr. Matthew Wallace's reports, he mentioned the following things to be done in the future. The focus control may need further improvement. The stereo audio decoder with Cyclone II FPGA is very noisy and needs improving or replacing (3.2.2). A groove tracking system must be designed and built.

3.2.1 The Read-Head

The read-head structure from Mr. T.J. Haworth [7] is summarized here.

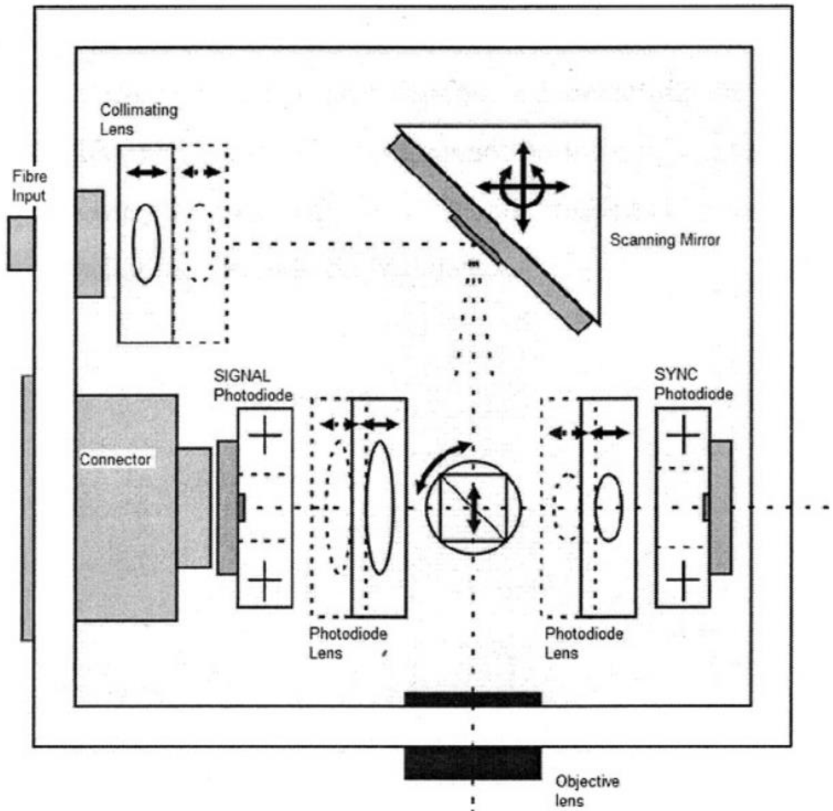


Figure 3.2.1 The layout of the optical read head (copied from Mr. T.J. Haworth's Report)

Figure 3.1 shows the layout of the read-head built by T.J. Haworth. a red laser beam is reflected off the surface of a $1mm^2$ micro-mirror, oscillating at 13.73kHz. The reflected light is split into two beams through the beam splitter. One goes to the 'SYNC' photodiode, which provides a trigger for scanning position, and the other goes to the objective lens. The objective lens focuses the beam to a 6um dot on the vinyl surface. The resultant system is thus a moving dot scanned across the width of the record groove (between 30μm and 80μm according to the RIAA standard [8]). The intensity of the reflected light is depended on the shape of the scanned position. For example, the reflected light will have higher intensity when the scanning dot is at land, and lower intensity when it is inside the groove. The reflected light is then passed back through the objective lens and the beam splitter to the SIGNAL photodiode where its intensity is detected.

This year, Professor Andrew S Holmes built a new improved read head:

- Improved optical design in read head. [21]
- Reduction of read head size. [21]
- Replacement of eccentricity and focus actuators with flexure-based designs. [21]
- Integration of high-voltage driver for MEMS mirror. [21]

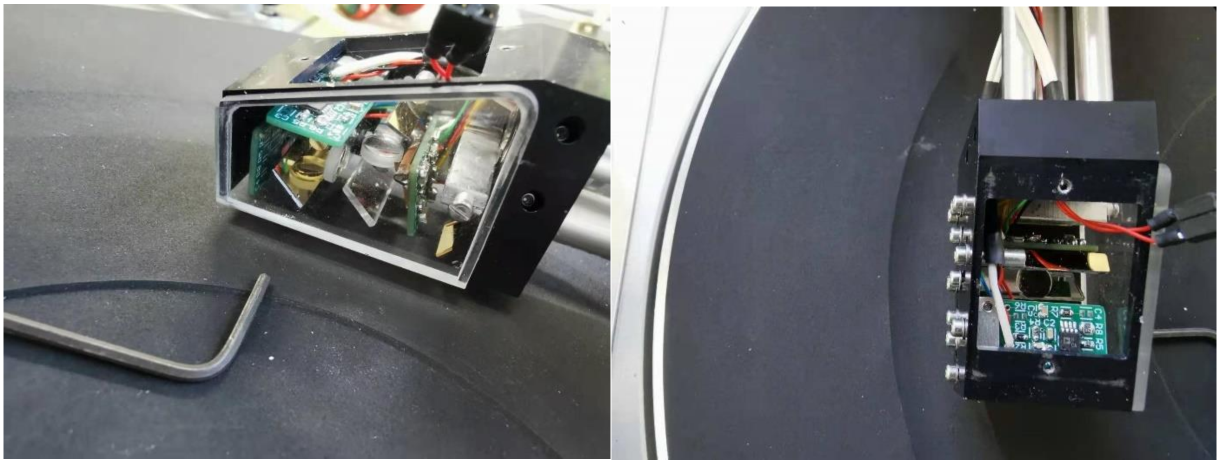


Figure 3.2.1b The side view and top view of the new read head. The oscillating mirror is on the top cover.

3.2.2. Audio Decoding Mechanisms

An audio decoding project on Cyclone II FPGA is completed by previous projects [4]; however, the result from generated groove data shows that this decoding results in some glitches in audio. The discussion in this section will thus focus on the algorithm and mechanism of converting reflected light intensity information into the audio wave, not the hardware. Then, a new hardware solution will be proposed in section 4.1.1 of the report.

a). Time to waveform method

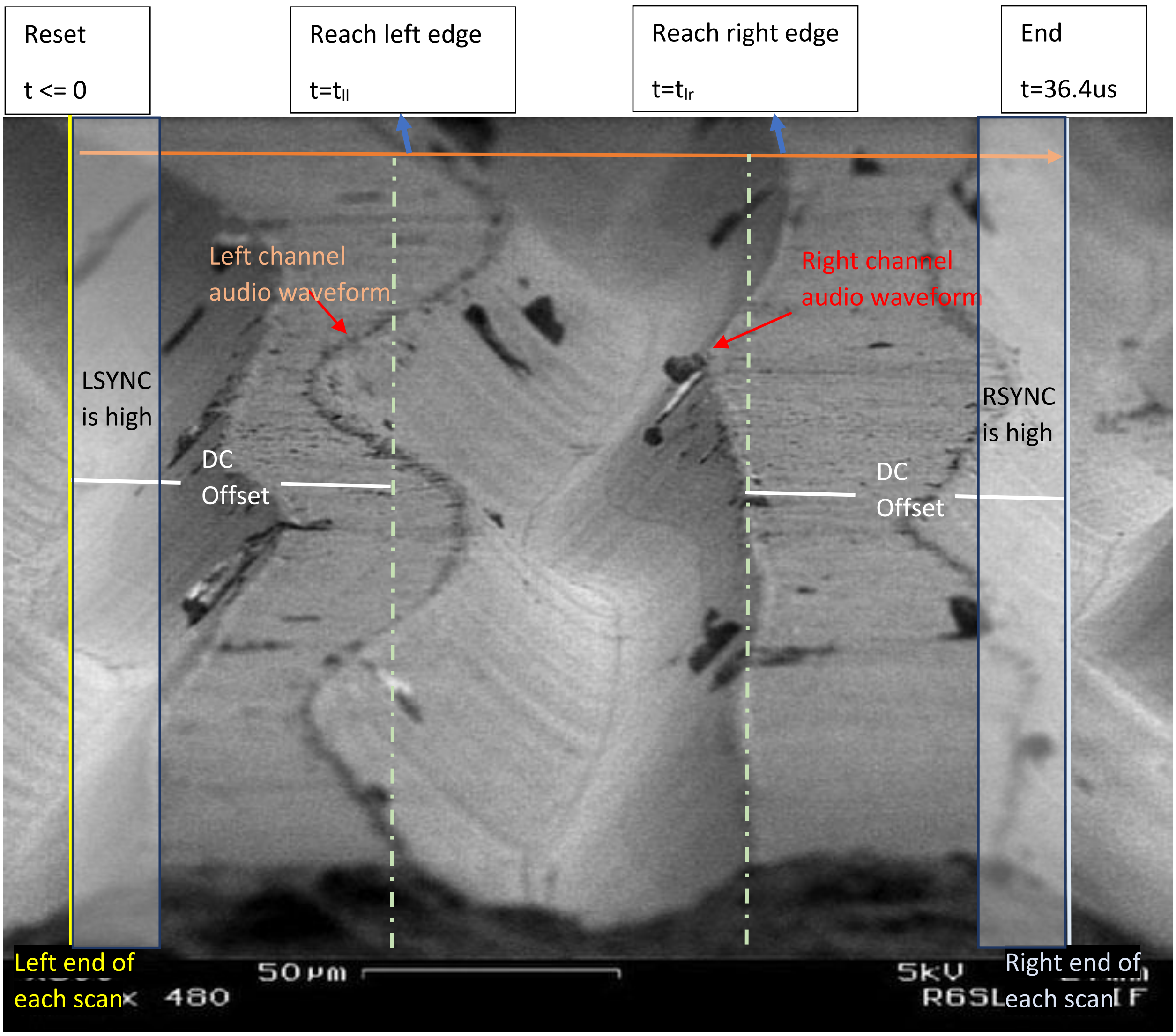


Figure 2.2.1.a. Image of groove taken by an electronic microscope.[9]

The Figure 3.2.1.a shows a detailed image of a groove of a vinyl disc. The yellow vertical line shows the left end of each scan, the blue vertical line shows the right end of each scan. The orange arrow shows one left-to-right scan as an example. At the start of each scan, the timer is resented to 0. After t_{ll} seconds, the scan reaches the end of the left land of the center groove. After t_{lr} seconds from the start, the scan reaches the edge of the right land of the groove. At 36.4us, this scan finish, and the next right to left scan will start. In the right to left scan, the same mechanism is used to acquire t_{rr} and t_{rl}. Then, all samples from the right-to-left scan will be transferred to left-to-right time coordinates by using the fixed relationship between the mirror oscillation and "SYNC" pulse. Now every t_{rl} and t_{rr} is transferred to new t_{ll} and t_{lr} .

There will be 26.5k pairs of samples t_{rl} and t_{rr} corresponding to left and right groove shapes (illustrated by the red dots in figure 2.2.1) per second. Since the groove shape is the waveform of the left and right channel audio signal, the t_{rl} and t_{rr} signals will also follow the shape of the waveform of the audio signal with some added DC offset (illustrated by the green dotted line). The DC offset of the left channel will equal the DC offset of the right channel if and only if the groove is at the center of the scanning range. Therefore, the difference between the DC offset of the two channels is used in a closed-loop control system to track the groove. The DC offset can be generated through the moving average of n samples.

The clock speed of the timer, numbers of samples for moving average, and tracking speed will be determined in section 3 after the specification of the product is determined.

b). Image-based reading

In [18], the author proposed another possible solution for reading vinyl groove audio data. Instead of using time intervals, the author uses a fixed interval scanning point to scan images of different areas of the vinyl disc. Then, the image is stitched together, and the audio is played directly from analyzing the color of the image near the edge of the groove. This method has the following pros and cons.

Pros:

1. Capable of having better error detection algorithms by utilizing modern image processing technics.
2. No mechanical groove tracking is needed, the groove tracking can be done inside the stitched image.

Cons:

1. Require higher resolution on photodiode and photodiode amplifier and DAC. The read head needs to be able to capture in-groove landscape, not just distinguish between land and groove.
2. Audio playback will have a delay. The audio playback cannot start until a full complete circular track is scanned and stitched.

3.2.3. Focus Tracking System

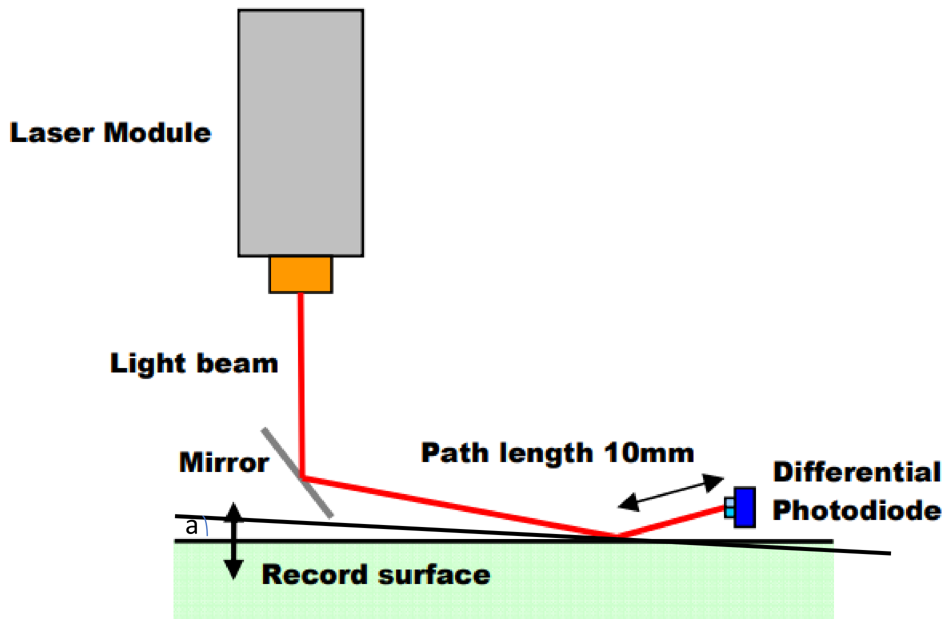


Figure 3.2.3. The structure of the tracking method applied to the project (copied from page 31 of [4])

The focus tracking system uses the position of the reflected laser beam to maintain the read head at 7mm above the record surface. A mirror reflects the beam to point to the disc surface with a big incidence angle. The incident beam is reflected by the surface of the vinyl. The reflected beam will point to the differential photodiode. Given that the angle ‘a’ between the record surface and the rotation direction is negligible. The reflected beam will only stay in the middle of the two photodiodes if and only if the record surface is 7mm underneath the read head. If the beam is not centered, the two photodiodes will receive light with different intensities. Such difference is converted to voltage and is amplified to drive the Focus voice coil motor. A closed-loop system is therefore implemented. The gain of the amplifier is tunable to make sure that the system can be stable and can have a quick settling time.

To minimize the effect of ‘a’, the incidence angle must be close to 90 degrees.

The error caused by ‘a’ can be canceled by using two such systems as shown below.

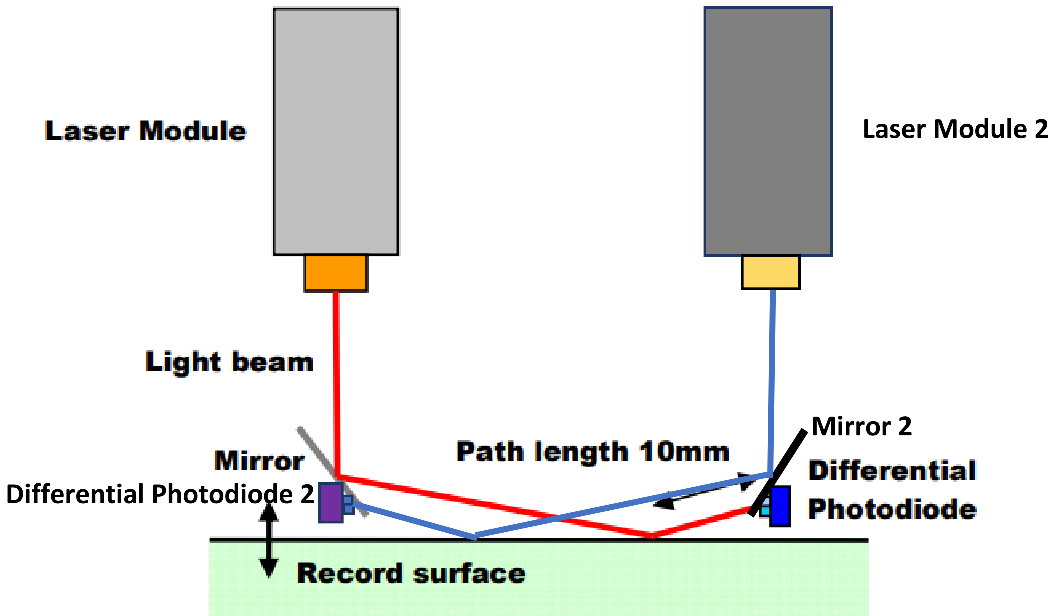


Figure 3.2.3.1 The structure of a better tracking method to cancel the effect of angle.

If a clockwise rotation is applied to the record surface, such rotation will cause the 'red' beam to shift downward and will cause the 'blue' beam to shift upward.

4. Project Specification and Requirement

4.1. Overview

The basic aim of this project is to design and build a working optical vinyl player. As required in section 1, the optical vinyl player should ideally be able to play any 33.3RPM record loaded by the user without the use of a stylus. The audio must be played in real-time (3.1.1). The sound quality should not be worse than a non-HIFI turntable. (3.1.2).

If the basic aim is achieved, there are more advanced challenges. First, the audio signal captured by time frame will need to be calibrated due to the nonlinearity of scanning movement. Second, error correction, including playing bad vinyl discs, 45rrpm and 78rrpm discs, should be implemented and improved. Moreover, other add-ons such as saving vinyl to wav files, connecting to a Bluetooth speaker, jumping between songs, etc. could be implemented. Finally, sound quality can be further improved.

To achieve those requirements, the following criteria must be met.

1. For all previously developed modules, there must be an improvement in this project.
2. Focus tracking should have a maximum error < 0.1mm.
3. Audio decoding should be able to handle 20kHz bandwidth stereo rather than 10kHz mono. The maximum SNR of the Audio decoding should be greater than 66dB. (This figure assumes that the scanner is noiseless)
4. The groove tracking should be able to track at least a full song without failure.
5. Sound linearity and frequency response should be improved by filters

6. The combined system should work as expected.

Since Vinyl discs are not designed to be played by optics. Playing it by lasers is very challenging. I may encounter lots of unexpected difficulties in the project.

4.1.1. The Details of Real-Time Playing.

It takes a CD player 5 to 8 seconds to prepare before it plays the CD. For this optical vinyl player, the time taken for the algorithm to focus and pick up the correct groove should not exceed 8 seconds. (This does not consider motor speed.)

When encountering a tracking error, the optical vinyl player should not stop playing. It could either extrapolate the previous sample or skip to the nearest available groove to continue playing.

Infinite looping of a part of the groove due to tracking error should be detected and skipped.

4.1.2. The Technical Spec of Sound Quality.

The sound quality needs to be specified by the **SNR** (signal to noise ratio) at the output, and frequency response. It turns out a trade-off is needed between max frequency response and max SNR. Also, another trade-off between control robustness and channel separation is needed.

The term **Dynamic range** is often used in this type of research. In the case of describing vinyl turntable performance, it is interchangeable with **SNR**. (Minimum usable signal strength = noise signal strength)

The theoretical maximum SNR of playing a vinyl record is 70dB. [10] Measurement of the SNR of turntables reveals that the actual output SNR lies between 60dB to 70dB. [11].

Therefore, the optical vinyl player should target an SNR at 70dB.

Since the position of each sample on the audio waveform is calculated from the time of the scan. (see section 2.2.2 for details). The minimum achievable noise power is the power of quantization noise due to the discrete property of the timer. Using the **6dB law**, reaching an SNR of 72dB needs 12 bits for each sample.

The scanning takes 36.4us to move across its scanning range. Assuming the scanning is at a constant speed. The maximum width of the center groove takes up half of the scanning range. 18.2us will therefore be converted into the left channel and right channel audio amplitude. That is 9.1us for each channel. Within this 9.1us, the counter must count from x to $x+2^{12}$ to provide 12bits audio data. The minimum frequency of the counter is thus given by 4096/9.1u=450.11MHz.

The audible frequency range is 20Hz to 20kHz. Most commercial non-HIFI non-portable speakers have a bandwidth from 45Hz to 20KHz. There is a challenge on both ends of the bandwidth.

If hardware is fixed, there is always a quantization noise and bandwidth trade-off issue at the high end. The current hardware (by the previous project) supports only 13.73kHz bandwidth (refer to section 2.2.2 where the sample rate of each channel is 26.5kHz). Suppose the bandwidth is extended to 20kHz. The sample rate is then 40KHz. The mirror oscillation frequency needs to be 20kHz. The scanning will thus take 25us to move across its scanning range. This leaves 6.25us for the counter for each channel. If the counter frequency is unchanged, the resultant resolution of each sample is reduced from 12bits to about 11bits.

As a result, doubling the bandwidth of the high ends of the spectrum will reduce the max SNR by 6dB at a given clock frequency.

For low-end frequency, there is always a tradeoff between groove tracking robustness and bass response.

Figure 3.1.2 shows a groove consisting of a pair of in-phase low-frequency signals put at the left and the right channel.

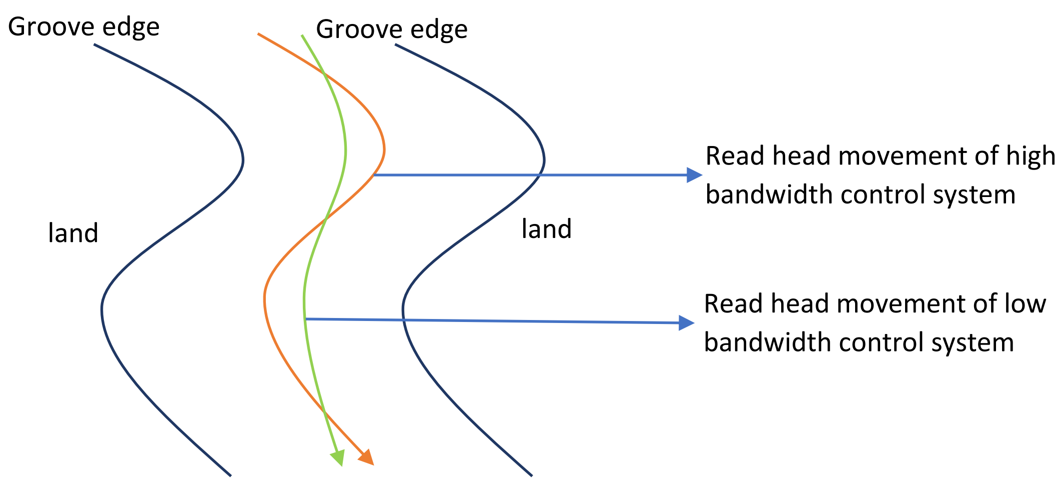


Figure 3.1.2. Low in-phase frequency and tracking.

The groove tracking system has its own bandwidth. If the bandwidth of the control system has a bandwidth not lower than the minimum reproduction frequency. The read head could move along with the audio waveform instead of picking up the waveform.

If the bandwidth of the control system is too low, it may fail to react to the movement of the groove position. The minimum bandwidth (about 87.2Hz under 1mm assumptions) of the control system is calculated in section 3.1.3. The bandwidth of the control system is set by the number of samples taken into the moving average DC offset generator mentioned in section 2.2.2.

Fortunately, the groove shape is proportional to the integral of the audio signal filtered by the RIAA curve. Though the RIAA curve suppresses low-frequency amplitude, the integral still made bass dominant in the groove. Traditional stylus pickups sense the movement of the stylus through magnetic flux-induced current which is effectively a groove differentiator. The optical read-out, however, picks up the groove shape without differentiation, allowing less corruption for the bass signal. This allows the control signal to corrupt some bass. The final system is tuned by Frequency sweep signals from “Cardas Burn” vinyl.

The final decision of the audio frequency bandwidth is estimated to be 45Hz to 20kHz.

4.1.3. Groove Tracking and Maximum Trackable Center Deviation

The main disturbance of the groove tracking system is the center deviation. The center deviation is defined as the distance between the rotation center and the physical center of the vinyl disc.

The situation is illustrated in figure 3.1.3.

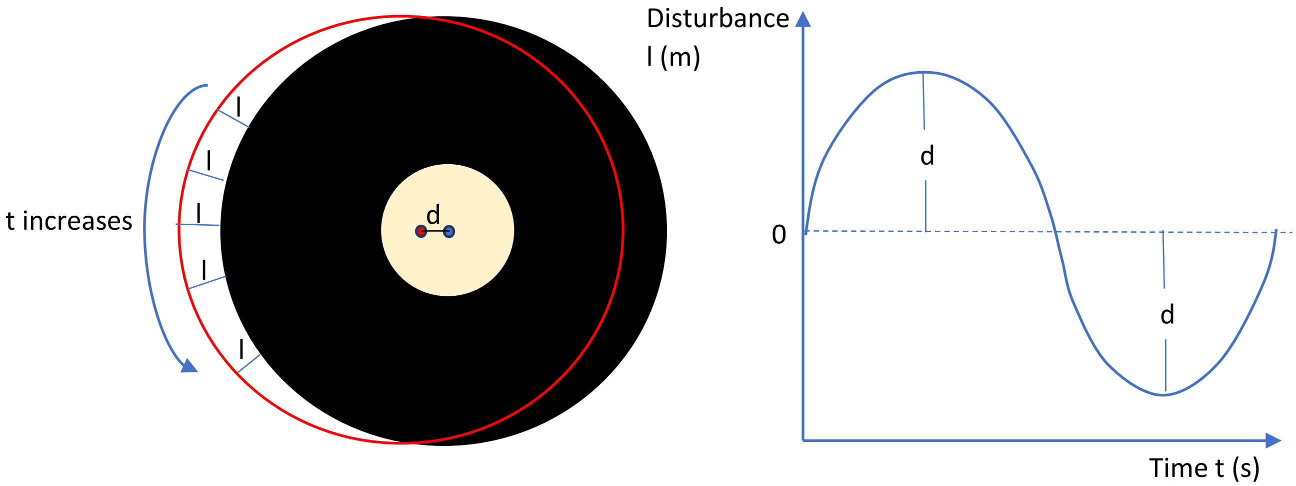


Figure 3.1.3. The center deviation of the vinyl.

Figure 3.1.3 illustrate the center deviation and the resultant disturbance. The physical center of vinyl (blue dot) is 'd' meters away from the rotation center (red dot). The 'l' indicates the disturbance caused by the center deviation 'd' while the disc is rotating.

$$l = dsin(\omega t)$$

The rotation speed of vinyl is 33.3RPM. $\omega = \frac{33.3}{60} \times 2\pi = 3.487\ rad/s$

$$l = dsin(3.487t)$$

To work out the bandwidth of the control system, one must figure out the rate of change of the disturbance.

$$\frac{dl}{dt} = 3.487dcos(3.487t)$$

According to the mechanism in section 2.2.2, to make sure the groove tracking is reliable, it is essential that the control system can move in either direction within every 'dt', and such 'dt' could not make $dl > \frac{w}{2}$, where w is the minimum spacing between two groove tracks center. Since the groove width is 30um to 80um depending on the amplitude of the audio signals, w = 80um.

$$\frac{w}{2dt} > 3.487dcos(3.487t)$$

The right-hand side will have a maximum value when t=0.

$\frac{w}{2dt} > 3.487d$ ➔ $dt < \frac{w}{6.974d}$

The bandwidth of the groove tracking system is approximated by $\frac{1}{dt}$.

To define the system's parameter 'd' effectively, one should measure the center deviation 'd' of at least 30 vinyl records. The 'd's will then follow a normal distribution. Then a d_{max}(maximum trackable center deviation) should be picked so that the probability of a randomly picked vinyl record having $d<d_{max}$ is greater than 99%.

Currently having more than 30 Vinyl records in hand is not practical, therefore an extreme value is picked. d_{max}=1mm. Therefore dt < 11.47ms. The minimum bandwidth of the groove tracking system is therefore 87.2Hz.

To figure out the maximum acceleration needed.

$$\frac{d^2l}{dt^2} = -12.159\ dsin(3.487t)$$

Maximum acceleration

$$\frac{d^2l}{dt^2}(max) = a(max) = 12.159\ d$$

Take d=1mm

$$a(max) = 0.0122m/s^2$$

The max weight allowed for the read head and voice coil motor is 1kg. $F = ma$

$$F = 1 * 0.0122 = 0.0122N$$

The current choice of voice coil motor can provide 15.6N force which is more than enough [13].

The maximum stroke of the current voice coil motor is ±3.18mm [13].

There are also some non-idealities of the voice coil motor to be considered.

1. The previous voice coil motor has a slide track, the slide track will have static frictional force >> dynamic friction force >> 0.0122N. The new flexure motor did solve this issue.
2. The system has a mass and a spring, such a system will have an oscillation frequency beyond which the movement and the input will have a phase different close to 180 degrees. This may result in positive feedback. The maximum bandwidth of the control system is also limited by this factor.

The modeling of such a system could get very complex, I plan to start with the basic linear model and then add those nonidealities to the model with the help of MATLAB simulation. The reading from [17] about various types of frictional force in servo systems and their compensation technology is useful in designing and tuning the groove tracking system.

5. Implementation

5.1. Module to be built and component choice

The module to be built and modified are:

Basic

1. A groove tracking system
2. Stereo audio decoding system with DC offset output for groove tracking
3. A more precise version of the focus tracking system
4. Central processing system that connected the above subsystem
5. Error correction algorithm

Advanced

6. Nonlinear audio correction algorithm
7. Noise reduction
8. Soundtracks pattern recognition for track skip, raw, etc.
9. Touch screen display and control for all functions

As all the modules to be developed are interconnected, it is, therefore, better to implement them on one beefy development board. This will also save cost.

5.1.1. Choice of MCUs/FPGA

As discussed in 3.1.2, to reach a max SNR of 66dB and frequency bandwidth of 20kHz, the required minimum clock frequency of the counter should be 450MHz. The Cyclone V 5EFA31C6 dev board available in the lab has a clock frequency of 50MHz for the FPGA part and a 900MHz clock for build-in cortex A9. The 50MHz available for dedicated hardware built-in FPGA is not sufficient (Though PLL can be used to multiply the clock frequency, the propagation delay does not allow a clock frequency > 200MHz). The 900MHz cannot be used outside of the cortex A9. Therefore, only the cortex A9 inside the Cyclone V is usable. Considering the cost of FPGA, this is not a good choice.

Originally, the audio DAC is also built on FPGA. Nowadays, a CD-quality commercial audio codec is cheap and widely available. It is, therefore, better to use this audio codec.

Audio codec SGTL5000 by NXP, a dual-channel 16 bits 44kHz I2S audio codec, is selected

Teensy 4.0 (NXP I.M.X RT1062), a development board with 600MHz arm cortex m7 processor and build-in I2S, is selected [14]. The 600MHz 32-bit CPU cycle counter can be used for precise time interval measurements.

The teensy 4.0 has a dedicated audio shield built with SGTL5000, 3.5mm audio jack, and SD card slot. This saves time on hardware development. Time is saved to develop more robust software and control algorithms.

Since an MCU is selected, the clock cycle counts of reaching the groove edge from the start of scans must be recorded through an interrupt. The interrupt latency will therefore be taken into consideration. The GPIO interrupt latency of NXP I.M.X RT1050 is 11 cycles, measured by the report of NXP [12]. This is true for all GPIO interrupts. This corresponds to 18.33ns and will not make two consecutive interrupts overlap. (Note that the RT1050 and RT1062 have the same processor.).

5.1.2. Choice of Motor driving signals

All motors are driven by PWM signals from the control loops.

In previous projects. All controllers are implemented by a P controller. A P controller in a spring-mass system failed to correct the steady-state error caused by the spring force. Therefore, this year, both groove tracking and focus tracking is implemented by PID controllers.

Implementing PID controllers using analog circuitry is not hard; however, tuning the bandwidth and gains of analog PID controllers within a large range requires changing hardware components and is time-consuming. Therefore, all PID controllers are implemented by MCUs digitally. This ease the tuning process in the development phase Those digital PID controllers, if noisy, can be replaced by analog equivalent controllers in the future when parameters are determined.

5.1.3. Choice of Other Components

This is the choice made by the previous projects.

Model number	Function	Rated voltage(V)	Key Spec
LA13-12-000(LTR)	Focus tracking	15V	Peak force = 15.6N Max stroke = 3.18mm
LA13-12-000(LTR)	Groove fine tracking	15V	Peak force = 15.6N Max stroke = 3.18mm
Zaber T-L SR150A [16]	Groove coarse tracking	12V	RS232 interface 6 Bytes or 0 – 5V voltage
MAX3232	TTL and RS232 convert	5V	Max baud rate 115200
OPA380	Read head Photodiode Amplifier	7V	GBP = 90MHz
TL072P	Other OPAMP circuitry	±3.5 to 15V(max)	GBP = 3MHz
L298N Dual Bridge	Voice coil motor driver	36V(max)	Max current = 3A
BPX48(F)	Photodiode	Vr=10V	Capacitance = 25pF

5.1.4. Choice of software

Teensyduino will be used to program the teensy 4.0 development board. This is open-source free software. The programing language is C, which is efficient.

MATLAB will be used to optimize audio output in advance features. The audio of frequency sweep decoded by the vinyl disc player will be recorded and sent to a computer. The frequency response will be analyzed. A filter is designed such that the filter minimizes the error of the audio for the optical vinyl player.

MATLAB will be used to model the control system if the basic linear module failed.

5.2. Hardware Implementation

5.2.1. Top view module connection block diagram

Note that not all pins are shown. All power supply connections are omitted. This part focus on digital and analog signal connections.

Blue arrows indicate electronic connections, black arrows indicate mechanical connections

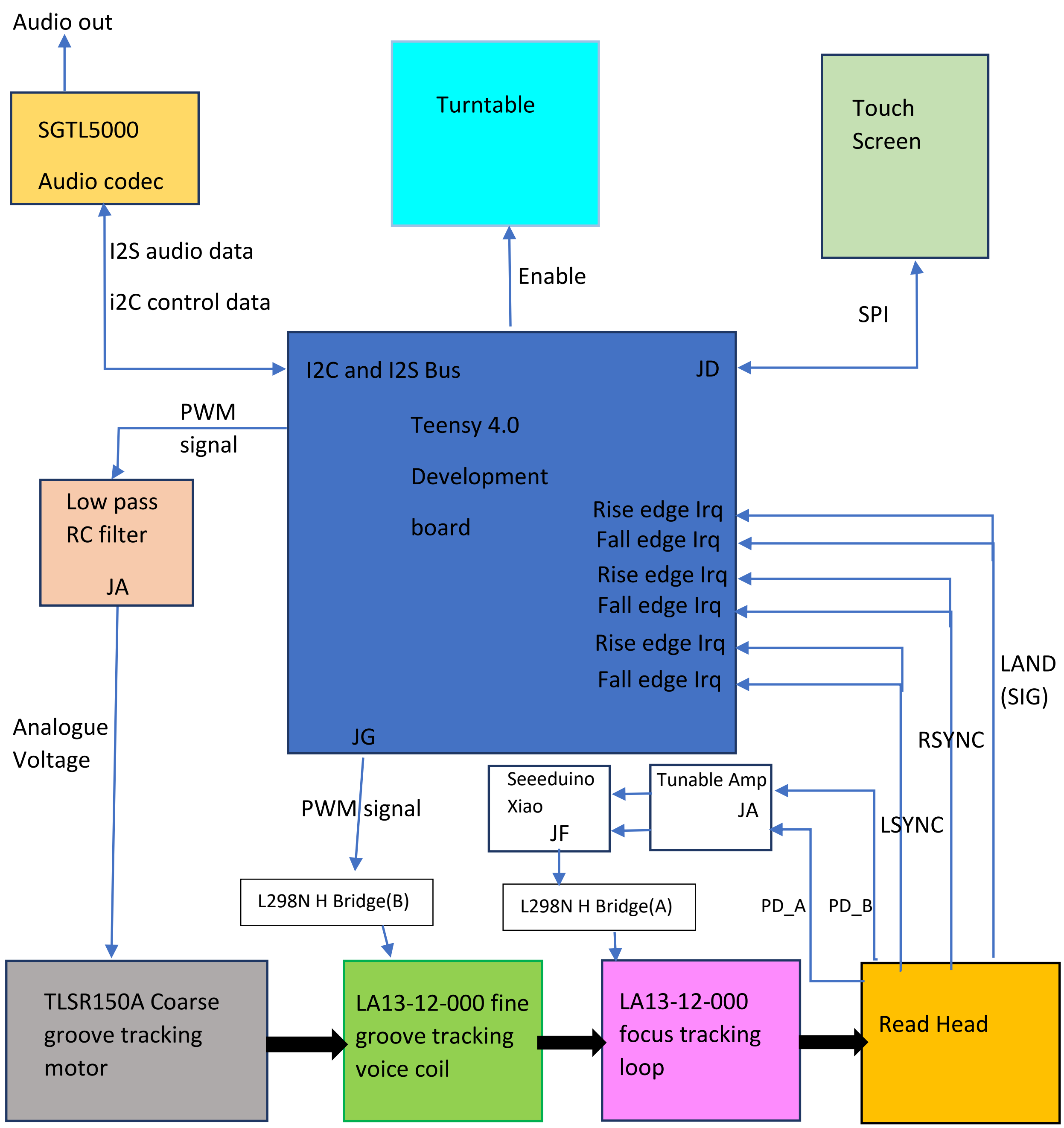
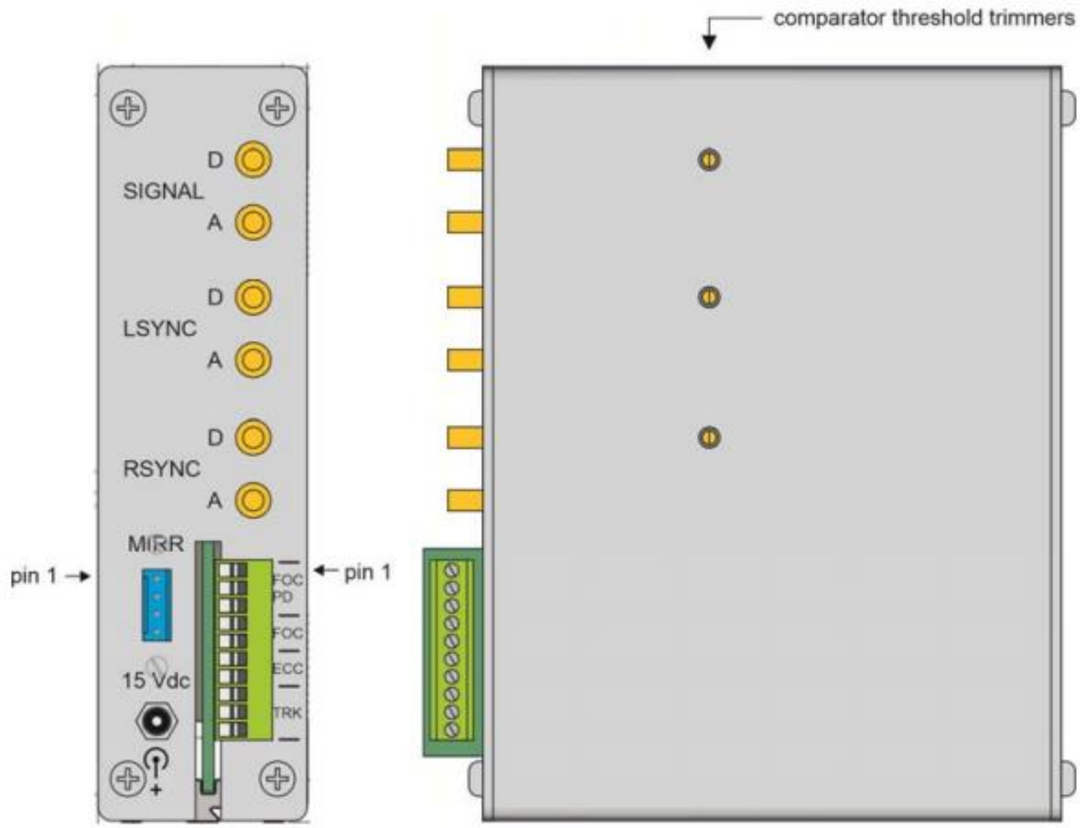
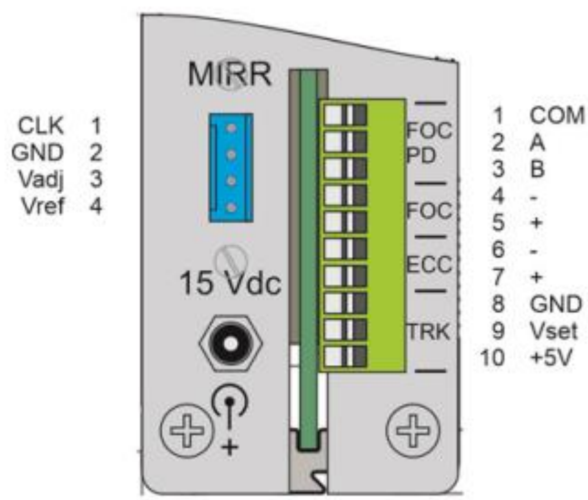


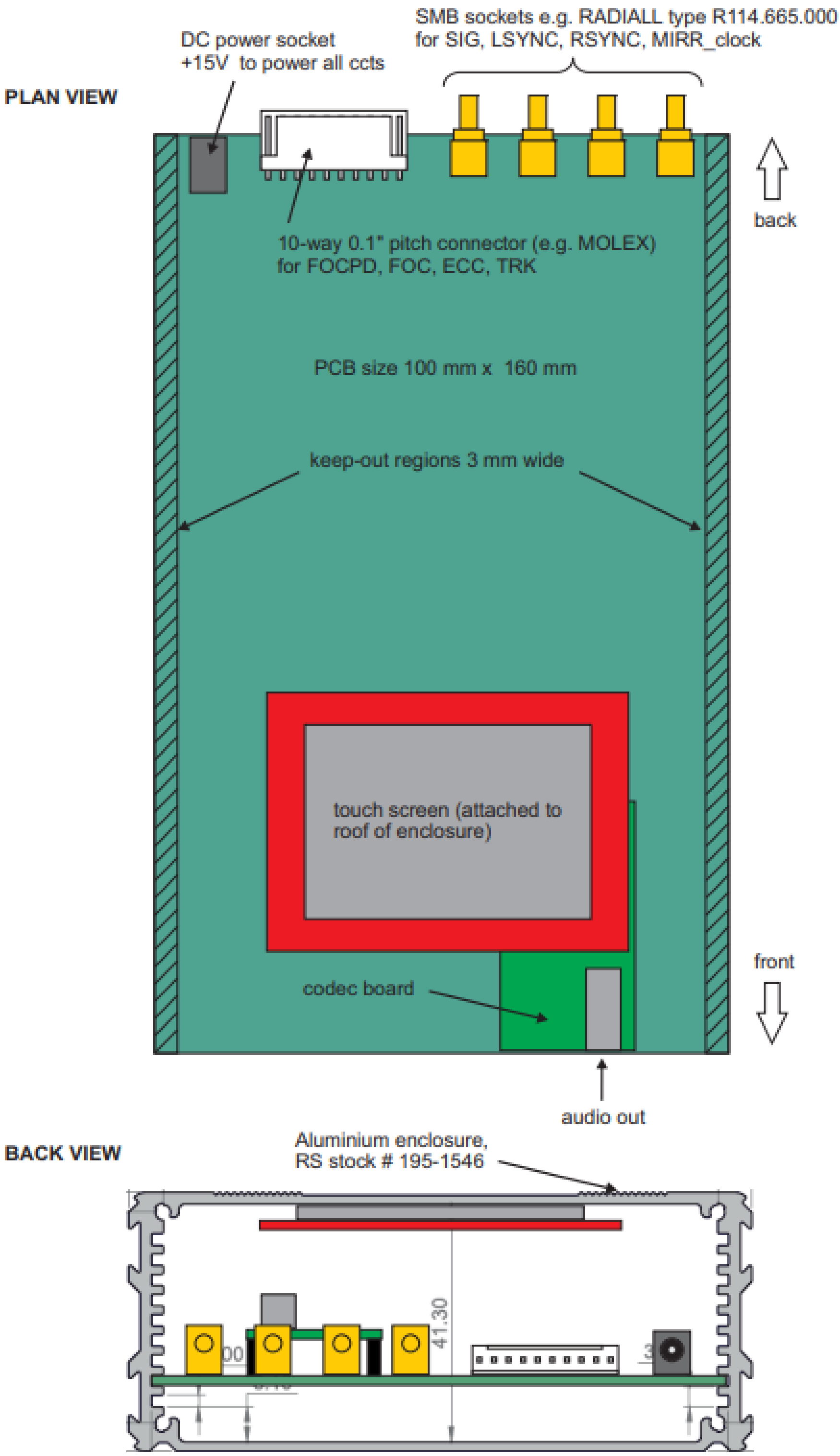
Figure 5.2. Module connection diagram

5.2.2. Hardware connections at read head [21]

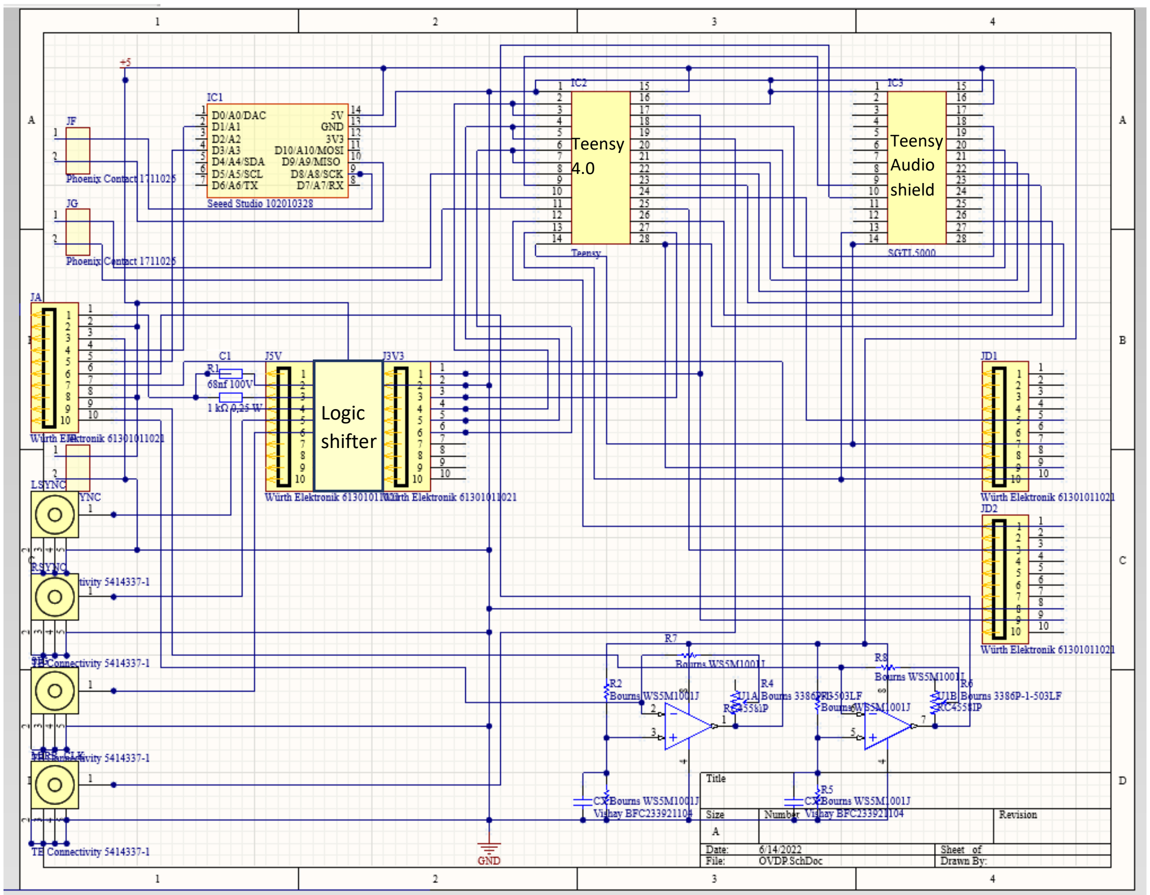


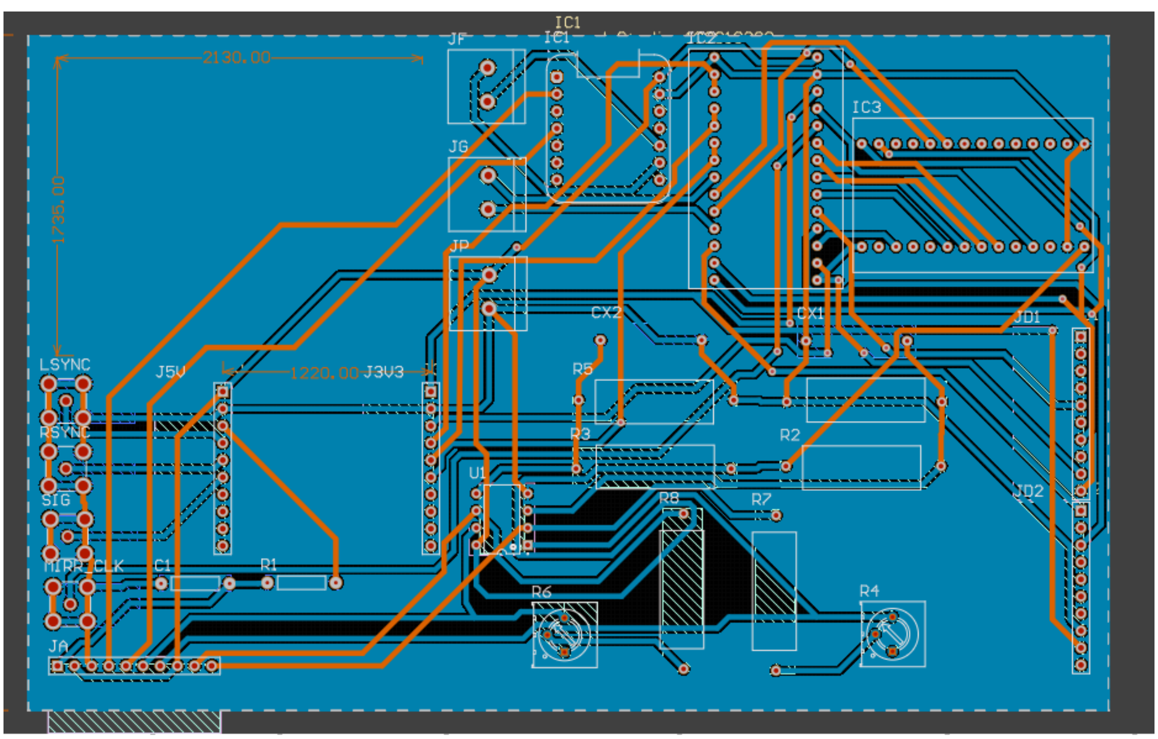


5.2.3. Hardware connection at Central processor [22]



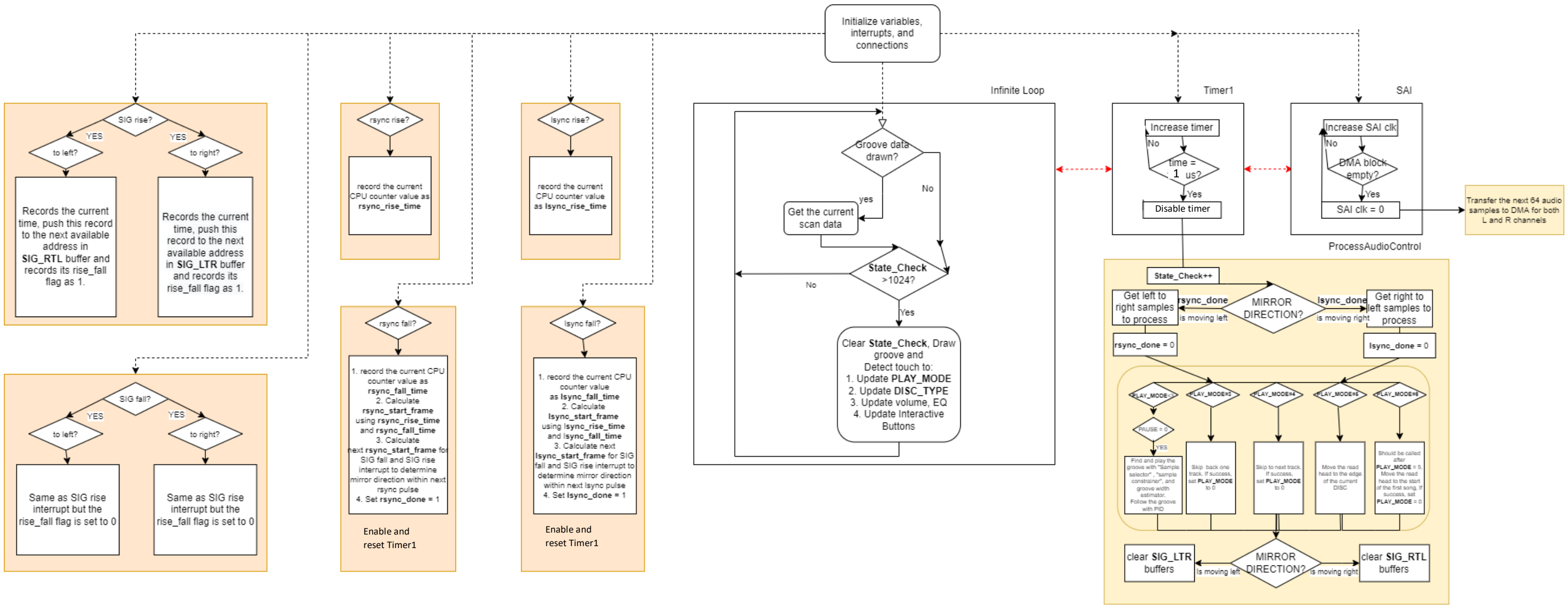
5.2.4. Full circuitry and PCB layout





5.3 Software Implementation and Block Testing

5.3.1. Top view software architecture



1. Doted red arrows represent simultaneous events. Black arrows represent sequential events

2. All colored boxes represent hardware interrupts. Their preempt priorities are such that ‘Orange’>’ Yellow’

3. Logics of each long rectangular box is concluded verbally. Their details are discussed in future sections.

5.3.2 Focus control system

The focus control system aims to maintain a constant distance between the read head and the disc. The system's performance should not be affected by the actual thickness of the vinyl disc. The accuracy of the system affects the higher-end response of the audio output. The focus tracking system should not create any abrupt movements when faulty, because all the rest of the devices are depending on the focus control. Since the current design has only one usable diode (Figure 5.3.2.2a), the design will assume that the disc surface under the read head is parallel to the horizon plane. The disc cannot be severely warped.

1. Selection of controller type.

The previous focus control system is implemented by a P controller. The driving voltage on the focus attenuator is proportional to the difference voltage (**e**) picked up by the two photo-diode (See section 2.3.3).

Assumes that the focus attenuator is linear (displacement is proportional to voltage **D**=**k*V**) and the gain of the system is $\mathbf{k_p}$**.** If displacement **Y** is required to get good focus (**e** = 0), the **P** controller will generate 0 volts at position Y; however, the motor requires **Y/k** volts for moving to **Y**. This suggests that a P controller will have a steady-state error when the displacement **Y** is not zero.

An integral controller is needed to resolve this issue. An **I** controller can generate a voltage proportional to the integral of all previous '**e**'. It is, therefore, capable of generating Y/k volts even when the current **e**=0.

The integral needs time to accumulate its values, therefore its transient response is slower than the P controller. It is, therefore, necessary to have both P and I controllers in the system, a D controller is also added to improve transient response.

The new Focus control system is implemented with a PID controller.

2. Choice of the error signal

This year, Professor Andrew S Holmes built an improved compact read head with integrated drivers. The reduction in weight and size of the read head made faster tracking possible. Smaller space reduced the available space for light paths. The top focus photodiode's light path is being cut by another component when the read head is closer than its best focus position (see figure 5.3.2.2). The positive feedback region from 1.8mm to 2.0mm made it impossible to use the differential voltage of the two photodiodes for tracking.

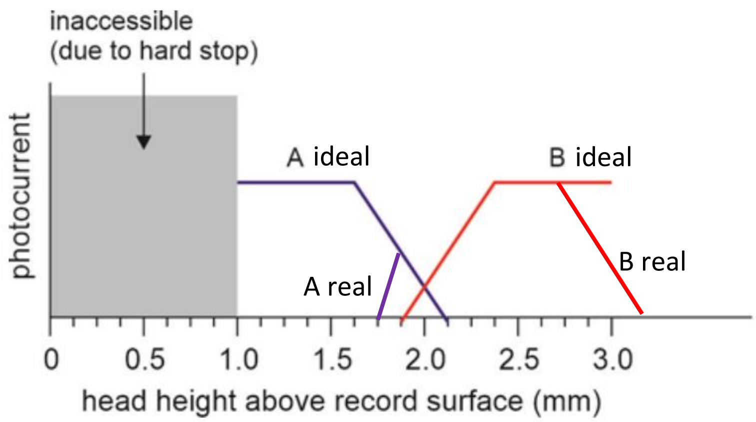


Figure 5.3.2.2a Voltage output of both focus pickup diode versus read head distance [21]

The error signal is therefore generated by only the button diode (B). The photodiodes are biased at 2.5V, their signals are amplified by the following inverting amplifier designed by Mr. Matthew Wallace.

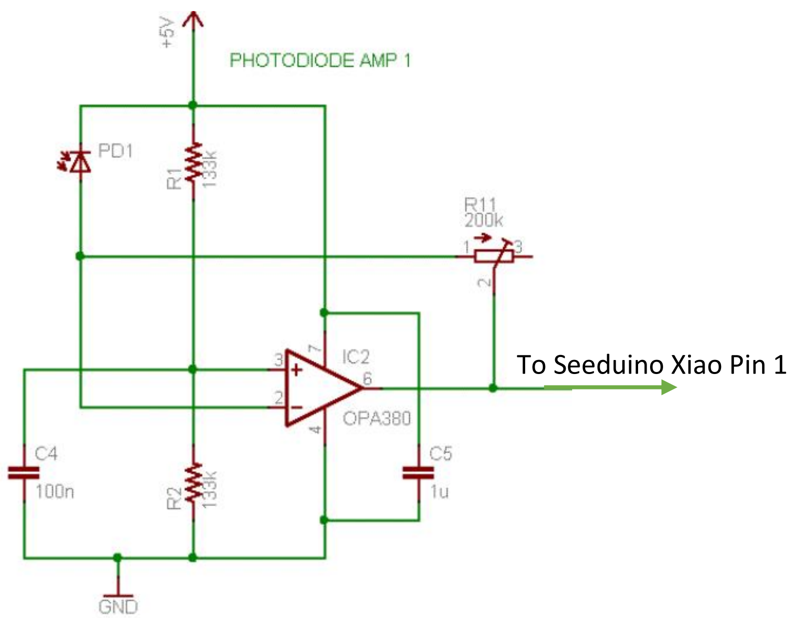


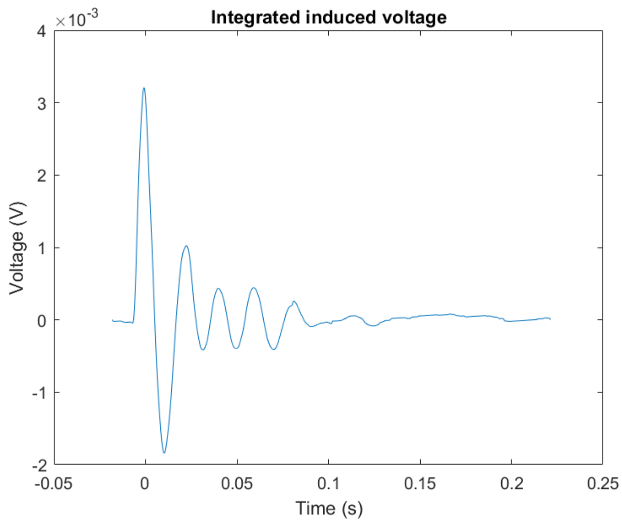
Figure 5.3.2.2b Inverting amplifier of focusing photodiode [4]

The amplified signal is read by a 10-bit 3.3V ADC on Seeeduino Xiao. The sample rate is 12kHz. ADC value is 450 when the read head is at best focus. The ADC value rises to 700 when the read head touches the disc. As read head moves away from the best focus point, the ADC value falls to 300 and then rises to 740. 450 – (ADC value) is the error of the digital PID controller. A positive error drives the attenuator downwards.

3. *The design of the focusing PID controller*

(1) Determine the maximum bandwidth of the system

The two-terminal of the attenuator is connected to an oscilloscope. The scope is set to single trigger mode. The attenuator is hit by hand to simulate a delta input. The waveform induced by the attenuator after the 'delta' is recorded by the scope. The waveform is integrated to estimate the shape of displacement transfer functions. The figure below shows the results. Note that the total displacement of the attenuator is zero throughout this test. The result of the integral should be offset correctly to make sure that it starts at 0 and ends at 0.



The oscillation frequency of the attenuator is 56Hz. Spring-mass systems have a 180-degree phase shift at oscillation frequency. This could result in positive feedback.
To make the system robust, a filter with a zero at 56Hz is needed. The cheapest filter is a moving average filter. Its *sinc* function in the frequency domain has a series of zeros at all harmony of a base frequency. This property is preferable when killing resonance.
One solution is to set 56Hz as the base frequency, i.e. 56Hz is the lowest zero of the filter. This design is not optimum due to two reasons. One, the resonance frequency is too close to the main lob of the sinc, a slight decrement in resonance frequency due to temperature could lead to instability. Two, the focus control system does not need a tracking speed as high as 56Hz.
A better solution is obtained by setting 28Hz as the base frequency. 56Hz is the zero between the second lob and third lob. The second lob is 13dB lower than the main lob and has 180 phase shifts. The system is, therefore, more robust against temperature rise. Better stability also allows a higher achievable gain and less error. 28Hz is also more than enough for focus tracking. Such filter is obtained by a moving average of 430 samples.
To obtain the transfer function of attenuator A(s) with correct amplitude, the parameter of the attenuator is needed.
The attenuator has an impedance of 23ohms.

$$Vout = Iout * 23$$

The attenuator has a BL value of 23.1N/A

$$Vout = \frac{Fout}{23.1} * 23 = Fout * 0.996$$

The attenuator has a spring constant of 15N/mm

$$Vout = Y * 15 * 0.996 = 14.9Y$$

As a result, A(0) = 1/14.9.

This means that the Laplace transform of the integral of the measured waveform should be multiplied by

a value such that it is 1/14.9 at s=0.

The Laplace transfer of the integrated induced voltage has a value of 6.484 x 10^{-5} at s=0.

The scaling value is therefore 1035. The following figure shows the transfer function A(jw) and a(t) of the focus tracking attenuator.

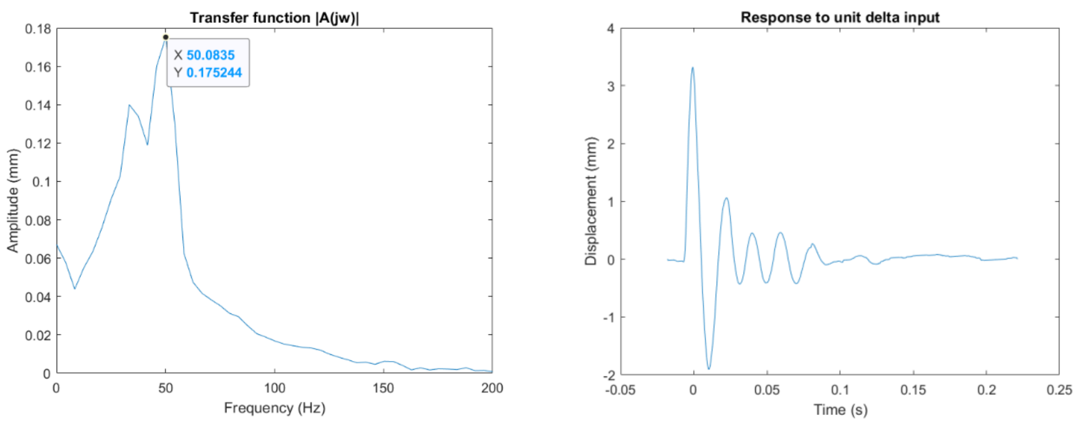


Figure 5.3.2.3a, the transfer function A(jw) and a(t) of the focus tracking attenuator.

Note that the measurement of the voice coil motor is a worst-case scenario. The analysis assumes that the driving circuit does not have electric damping to the voice coil. This is not true. The electric damping of the voice coil is dependent on the circuitry. The output impedance of the circuit used in this design is around 1 ohm. More damping means less resonance.

(2) Modeling and Design theoretical **kp**, **ki**, and **kd**

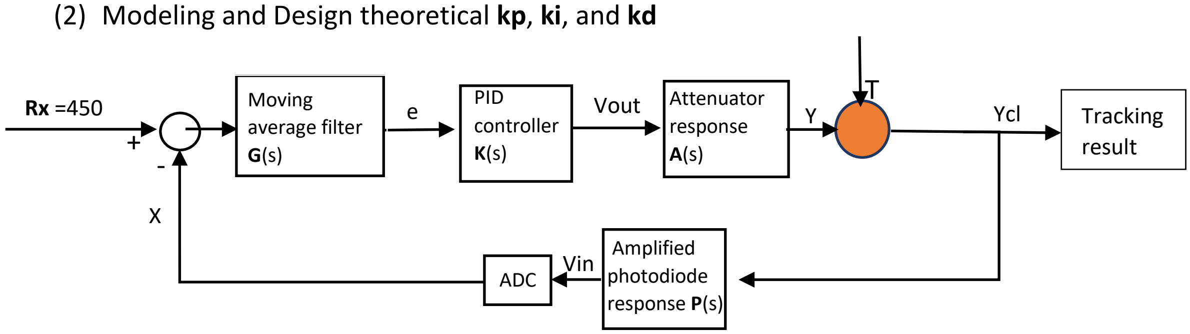


Figure 4.3.2.3b. The model of the focus control loop.

In the design phase, it is reasonable to assume the **P**(s) is ideal. Let P(s) be a constant **P**.
Let ADC be a constant **D**. Then,

$$X = \mathbf{PD}Ycl$$

Let Y be the displacement of the attenuator in 'mm'. Let **Ycl** be the distance of the read head to the disc. Let X be readout out of ADC. **PD** is measured empirically to be 400.
The close loop response is given by:

$$\boldsymbol{Ycl} = \frac{\mathbf{K}(s)\mathbf{A}(s)\mathbf{G}(s)\boldsymbol{Rx} + \boldsymbol{T}}{1 + 400 * (\mathbf{K}(s)\mathbf{A}(s)\mathbf{G}(s))}$$

Where $\boldsymbol{Rx}$ is used to set the required focusing position. The goal is to design a control system such that the **Ycl** does not change with **T** (record thickness).

The moving average filter $\mathbf{G}(s)$ is determined to be a 430 moving average filter, its transfer function is given by:

$$\mathbf{G}(s) = \frac{1}{430}\frac{1 - e^{-\frac{430}{12000}s}}{1 - e^{-\frac{1}{12000}s}}$$

The **K**(s) is the transfer function of the PID controller, its transfer function is given by:

$$\mathbf{K}(s) = K_p + \frac{K_i}{s} + K_d s$$

Then,

$$\boldsymbol{Ycl} = \frac{(K_p\mathbf{A}(s)\mathbf{G}(s)s + K_i\mathbf{A}(s)\mathbf{G}(s) + K_d s^2\mathbf{A}(s)\mathbf{G}(s))\boldsymbol{Rx} + \boldsymbol{T}s}{s + 400\mathbf{G}(s)(K_p\mathbf{A}(s)s + K_i\mathbf{A}(s) + K_d s^2\mathbf{A}(s))}$$

Note that **T** is multiplied by *s*, let *s*=jω. As ω (angular frequency) tends to zero in steady-state, the term **T**jω tends to zero while other terms don't. This confirms that the system has no steady-state error. The DC gain is given by setting s=0.

$$\boldsymbol{Ycl}_{ss} = \frac{K_i\mathbf{A}(s)\mathbf{G}(s)}{400\mathbf{G}(s)K_i\mathbf{A}(s)}\boldsymbol{Rx}$$

$$\boldsymbol{Ycl}_{ss} = \frac{\boldsymbol{Rx}}{400} = \frac{\boldsymbol{Rx}}{\boldsymbol{PD}}$$

Since $X = \mathbf{PD}Ycl$, $X_{ss} = \mathbf{Rx}.$ Steady-state error e_{ss} = 0.

$\mathbf{A}(s)$ is Laplace transform of the impulse response of the attenuator measured in the previous section.

The transient response of the **Y**cl can now be simulated numerically in MATLAB against turbulence **T** under different K_p, K_i, and K_d. The MATLAB did provide PID tunning toolbox for designing PIDs; however, that toolbox required a mathematic model for the open-loop response, The mathematic models in that toolbox failed to model the resonance of the spring. As a result, I write my own code to simulate the system numerically using FFT and Inverse FFT. The result is shown in figure 5.3.2.3b.

The system is simulated at K_p=0.03, K_i=0.01, and K_d=0.001. **Rx** is initialized at 550. A 0.03s pulse at 0.045s is injected to **Rx.** It takes around 0.0005s for the tracking output to vary with **Rx**. **T** is initialized at 1mm. A 0.03s pulse is injected at 0.125s to **T** to simulate a change in Disc height. The system quickly adjusts itself to the correct distance 1.375mm to adapt to the change in around 0.0004s.

The tracking results look very glitchy. This could be due to poor accuracy and delay in the measurement of the impulse response of the attenuator. When implemented the controller on hardware, the controller is stable.

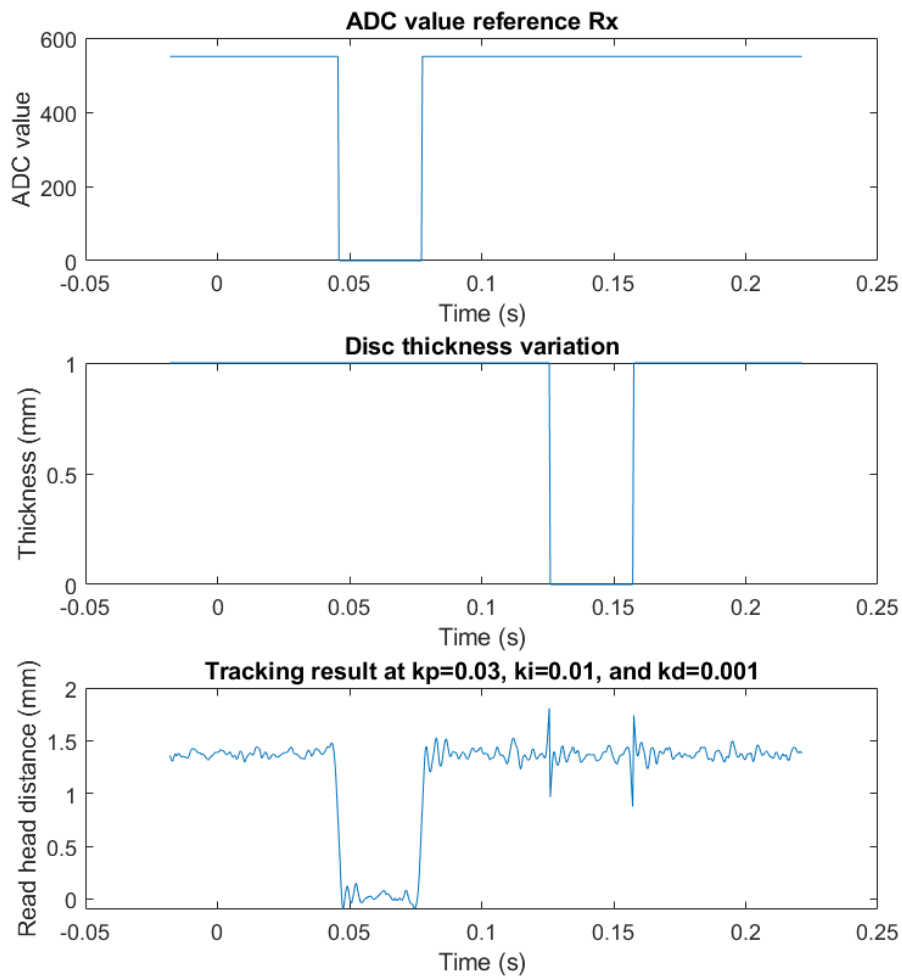


figure 5.3.2.3b. The close loop tracking performance against input **Rx** and turbulence **T**.

(3) Map the controller to the hardware

To implement the controller on hardware, the parameters should be transferred to digital values. The mapping is based on steady-state and A(s) is considered to be a constant.

The *Dout* of the PID is a PWM value from -1024 to 1024, this maps to -15V to +15V *Vout*.

$$Dout = Vout * 68.4$$
$$Vout = 14.9Y$$

To change the K_p, K_i, and K_d, from voltage domain to digital domain D_p, D_i, and D_d, 68.4 should be multiplied to all of them.

In the actual design, the moving average filter is placed after the error 'e'. Also, the moving average filter is designed to have a gain of 430 so that no division is needed. As a result, K_p and K_i parameter should also be divided by 430 to accommodate this change. This modification does not affect system performance.

After testing and adjusting on hardware with a warped (1.5mm) 45rrpm 7 inches vinyl, the final decision of D_p, D_i, and D_d is $\frac{5}{8192}$, $\frac{3}{2048}$, and $\frac{25}{4096}$. The specific denominators are to make sure that divisions can be done by simple right shift.

(4) Dealing with error and fault.
The throw of the voice coil, the pickup range of the photodiode could run out. This could lead to an overflow of integral. Once the integral overflow, a big jump will occur on the focus attenuator, and this will disrupt the groove tracking.
To avoid integral overflow, the following logic is designed. The logic is shown in pseudo-code.

```
If the tracking force is not at the limit
{
  Run the integral
}
If the tracking is at a high limit
{if(the error is negative)
  {
   Run the integral
  }
 Else
  {
   Do nothing
  }
}
If the tracking is at a low limit
{if(the error is negative)
  {
   Do nothing
  }
 Else
  {
   Run the integral
  }
}
```

The different disc has different reflectivity. When there is not enough light reflected to the read head, the read head should always move to the upmost position to reduce the risk of damaging the disc. This function is already inherited in the design because only the button photodiode is used.

(5) Testing the focus control
There is no accurate way to measure the distance. However, the position of the focusing light dot on the vinyl related to the read head should be the same if the focus control functions correctly. This can be verified by eye.
The test is carried out on 7-inch vinyl, 140g 12-inch vinyl, and 220g 12-inch vinyl. For both 7-inch vinyl and 140g vinyl. The test passes. For 220g vinyl, the PWM of the output reaches its max and the focus control does not function correctly. If the turntable is lowered, the 220g vinyl case

passes, and the 7 inches cases failed with the read head staying at the upper position. This is because the 7-inch disc is too far from the read head and there is not enough light intensity for the control.

5.3.3 Audio sample pick-up algorithm

The audio sample pick-up is an essential part of this project. This system need not only to correctly reproduce raw audio from scanning results, but it also needs to provide information for the groove tracking system. This means:

1. The audio sample pick-up system should pick the same groove regardless of the error of the groove tracking system, as *long* as the desired groove is within scan range.
2. The audio sample pick-up system should not use the pre-scan method to capture the groove width. It should capture the groove width in real-time.
3. The audio pickup system should not pick the right channel of one groove and the left channel of another (the next turns). If this happens, it needs to detect and correct it by itself.
4. If there is dust within a groove, as long as the size of the dust is reasonably small, the algorithm should not detect and correct Itself (as in 3) to the dust.
5. If the groove is not 'visible', it needs to roughly predict the groove so that the groove tracking system still functions.
6. It needs to remove glitches that do not belong to the music through slew rate and amplitude limiting, and this should not cause audible compression to the music.
7. It needs to take logs about the number of missing samples so that the groove tracking control system knows when to start or stop the coarse tracking motor.

The audio pick-up system developed by the previous project is a scan-pick-discard design. This means that the system starts scanning and waits for a satisfying sample. When the sample comes, the system stops recording and outputs the sample to the audio and control system. The system discards all other samples within each scan. It can only pick up the left-most sample near lsync. This does not meet the criteria.

To meet all criteria, the audio pick-up system must be a scan-select-constrain-pick-discard design. The data structure and timing diagram are shown in figure 5.3.3.1b.

The figure 5.3.3.1a shows the data flow of the whole audio pick-up system

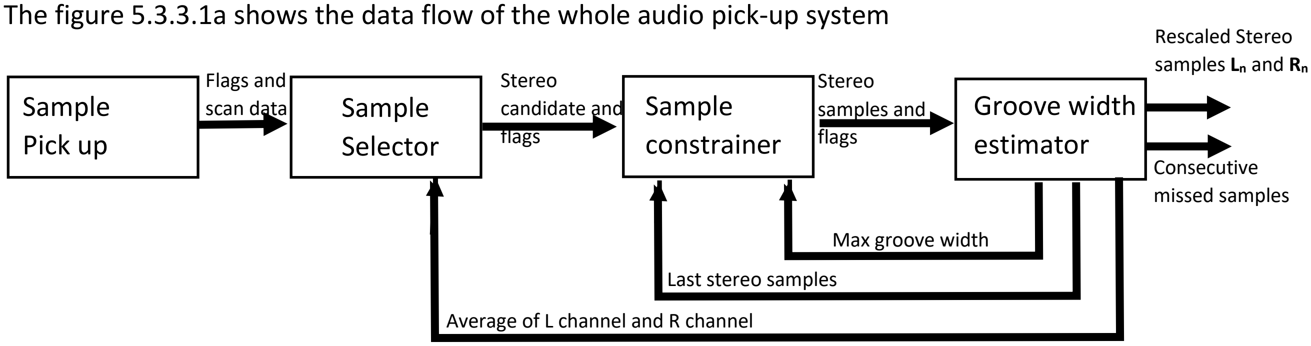


figure 4.3.3.1a data flow of the whole audio pick-up system

5.3.3.1 Sample measuring interrupts

The sample measuring interrupts aim to precisely record the time of all edges of SIG, LSYNC, and RSYNC to buffers for the rest of the system. The time is measured by the value of the 600MHz CPU cycle counter.

1. Data structure

To make a scan-select-constrain-pick-discard design, each scan can only be processed when they are finished. While they are being processed, the new scan generates new edges to be recorded. This means that the system needs to process the last scan and record the new scan at the same time. An alternative A-B buffered data structure is needed to achieve this.

Since the scanner consists alternatively of left-to-right and right-to-left scans, I decided to associate all left-to-right scans to A and all right-to-left scans to B. The switching between these two buffers will then be controlled by hardware signal lsync and rsync. This design is the most efficient approach but it did require both lsync and rsync to present.

If only one sync is available, we would need a big buffer for a pair of scans, then another buffer for the next pair. We would then split each buffer into two by comparing its elements to the middle time of each frame. This is a backup plan for the project if the read head can only provide one stable sync.

It turns out that the read head can generate reliable lsync and rsync. Also, the backup plan requires a complete change in the data structure. Also, the scan movement is not perfectly symmetric depending on drive force, temperature, and mechanical warm-up. (The closer the available SYNC to the selected SIG, the better the sound quality). Considering the complexity and low priority, the backup plan will not be implemented.

The data structure and timing diagram are shown in figure 5.3.3.1b

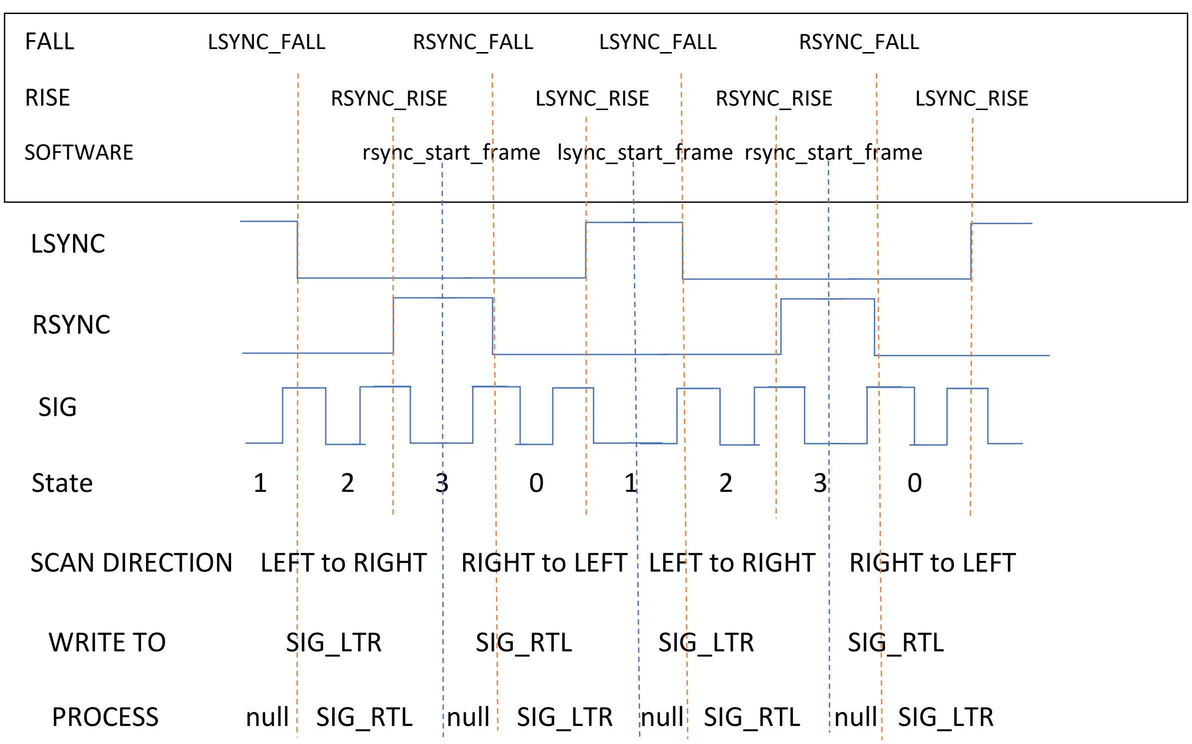


figure 5.3.3.1b The data structure and timing of the software system

2. Details of the pick-up interrupts

As indicated in 5.3.1, there are six interrupts for recording LSYNC, RSYNC, and SIG. Each signal drives two interrupts, one rise edge, and one fall edge. To make the recorded value precise, each interrupt will be kept as short as possible. Overlapping of two edges will still cause errors in recorded value, a negative feedback frequency lock loop algorithm is developed to remove this problem for LSYNC and RSYNC in the audio enhancer.

For a pair of scans, we have four states.

0: the scanner is moving to left 1: LSYNC rise 2: the scanner is moving to right 3: RSYNC rise

All states can be updated by SYNCs.

LSYNC_rise interrupt will set the state to 1. LSYNC_fall interrupt will set the state to 2. RSYNC_rise interrupt will set the state to 3. RSYNC_rise interrupt will set the state to 0.

At states 1 or 3, the SIGs could also update the states to 2 or 0. If it is certain that the scanner is passing its left-most or right-most points. This is implemented as follows.

Assuming that the scanner is now moving from left to right. In LSYNC_fall interrupt, the system will calculate the start time of the current scan using lsync_start_frame = (lsync_rise_time + (lsync_fall_time-lsync_rise_time)>>1) to predict the start of the current ongoing LTR scan. It will also predict the start of the next RTL scan using the start time of the last RTL scan and the start time of the current LTR scan split_sync = (rsync_start_frame + rsync_start_frame-lsync_start_frame). When the scanner moves to the point making RSYNC rise, the states become 3. Now both SIG_fall and SIG_rise interrupts will start to compare times of every upcoming edge with split_sync, as long as one edge arrives after split_sync. It sets states to 0 and switches to writing RTL buffer. The same mechanism is used for the right-to-left scan.

If scan frequency is slow, it is better to disable SIG from writing states to improve sound quality. Making SIG update states to 0 or 2 before the SYNC update will reduce the width of the 'null' region in PROCESS in figure 5.3.3.1a and leave more time to process. The drawback is that within the null region, the new _start_frame is not available. The new _start_fame is the end frame of the last frame that is under processing. Using an end frame value predicted from the start of the last frame results in more noise due to the temperature dependence asymmetry of the scanner. The prediction of future _start_frame should only be used for buffer switching within the SYNC pulse.

The SIG_rise and SIG_fall interrupt has the same structure. The only difference is that the sample recorded by SIG_rise will have a rise_flag = 1. The sample recorded by SIG_fall will have a flag = 0. Within each of these SIG interrupts, there are four possible cases determined by the states when a SIG edge comes.

0: The interrupt pushes the new time value to the next available position in the SIG_RTL buffer and records its rise_flag to SIG_RTL_r.

1: If the time is smaller than split_sync predicted by the last RSYNC_fall interrupt, push it to the next available position in SIG_RTL. Otherwise, it clears the SIG_LTR buffer if this is the first sample after this split_sync, then records all upcoming samples to the SIG_LTR buffer.

2: The interrupt pushes the new time value to the next available position in the SIG_LTR buffer and records its rise_flag to SIG_RTL_r.

3: If the time is smaller than split_sync predicted by the last LSYNC_fall interrupt, push it to the next available position in SIG_LTR. Otherwise, it clears the SIG_RTL buffer if this is the first sample after this split_sync, then records all upcoming samples to the SIG_RTL buffer.

5.3.3.2 Sample selector

A scan may contain one to five pairs of stereo data. The sample selector aims to pick up one pair of stereo data that belongs to the same turn of the groove as the previous pick.

In all the rest of the report, all "selected left channel samples" refer to the time difference between the actual time of arrival of the selected sample and lsync_start_frame. All "selected right channel samples" refer to the time difference between the actual time of arrival of the selected sample and rsync_start_frame. Their values of them are always positive.

According to the logic of pick-up interrupts, the following conclusion can be made about the raw scan data.

1. They are placed in time-increasing order.
2. All samples that belong to the left to right scans are placed in the SIG_LTR buffer, and all samples that belong to the right to left scans are placed in the SIG_RTL buffer. The pointers to these buffers always point to their next available positions. i.e., the total number of samples in each scan is equal to the pointer value minus one
3. At state=0, all samples, as well as the start time and end time of the last left to right scan, are measured. At state=2, all samples as well as the start time and end time of the last right to left scan are measured.

According to these conclusions, the sample selector can be built. The sample selector process left to right scan if it detects new rsync_done and state = 0. The sample selector process right to left scan if it detects new lsync_done and state = 2. This is also indicated in figure 5.3.3.1b.

For all pairs of left-right channel data, the left channel must be on the right of the right channel data. For all SIG_LTR, left channel data must be a rising edge, the right channel must be a falling edge, and the left channel data must have a higher position index in SIG_LTR. For all SIG_RTL, left channel data must be a falling edge, the right channel must be a rising edge, and the right channel data must have a higher position index in SIG_LTR.

The following diagram demonstrates the black box view of processing SIG_LTR, the design for SIG_RTL is the same except for the difference discussed in the previous paragraph.

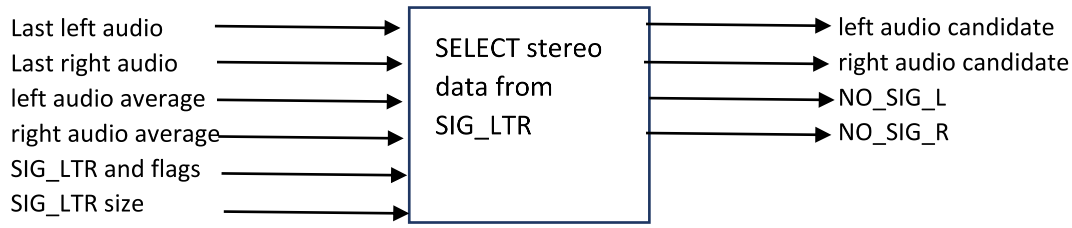


Figure 5.3.3.2 top view of the SIG_LTR sample selector.

To optimize the selected left and right audio candidate, a prediction needs to be carried out before the selection process. The prediction algorithm must be low-cost and fast. The prediction algorithm should work with any audio content.

Two models can be used.

1. Micro-model: Since low-frequency amplitude is higher in most sound and audio content in vinyl is band limited. It can be assumed that the next sample is equal to the previous sample if the time between them is infinite short. If the scan rate is fast, this assumption is valid.
2. Macro-model: The audio on vinyl has no DC component. The audio on vinyl can be any sound. As long as the sample size of these sounds is big enough, the audio sample from these sounds will follow a zero-mean normal distribution.

According to Micro-model, the best candidate for the left channel sample should be the one that is closest to the last left channel sample. The same applies to the right channel. According to Macro-model, the best candidate for the left channel sample should be the one that is closest to the DC offset ($\mathbf{DC_L}$) of the left channel. The DC offset is estimated through a moving average filter of size 1024. The same applies to the right channel.

After testing on hardware (after the development of groove tracking control), it turns out that Macro-model gives the most stable results in the sample selector. This could be due to the low scanning rate of the scanner. Also, Micro-model is more likely to cause skipping due to the lack of central pulling property. However, Micro-model is suitable for the ‘sample constrainer’ for removing glitches after the sample selector.

If the selector selects a real sample, it will write NO_SIG to zero. Otherwise, it will output the last sample and write NO_SIG to 1. Both channels follow this rule.

4.3.3.3 Sample constrainer

The stereo candidates selected by the sample selector may not be correct. The sample constrainer verifies these selections further. If the selection is not correct, the sample constrainer will correct them or estimate them by the last samples. The sample constrainer should be tuned with audio sources with different loudness from different discs to ensure that it does not cause extra audible compression or distortion to audio.

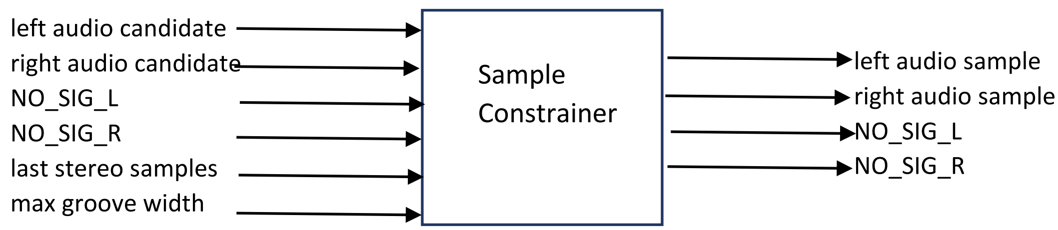


Figure 5.3.3.3 top view of sample constrainer.

1. **Slew-rate constrain**
   Each groove contains both right channels and left channel data. Assuming that the average groove position and max groove width (**G**) are constant between two consecutive samples, each channel's signal's jump (**J**) within two consecutive samples can only take half of the max groove width. Therefore, the slew rate is limited to 0.5**G** (*clk* * fs). **G** is estimated by groove width estimator measured by the number of CPU counter (*clk*). For shellac disc single edge playing mode, the slew rate is hard coded as SCAN_RANGE/6 (*clk* * fs)
2. **Left-right channel sample relative position constrain.**
   The distance measured in *clk* of the left and right channel candidates can not exceed the predicted max groove width (**G**).
3. **The sample constrainer can be turned off by the serial terminal via a Boolean variable.**
   This is to allow instant comparison of audio quality.

If the new candidate satisfied the two constrain, output it to the next block and set NO_SIG_ to 0. Otherwise, output the last available sample to the next block and set NO_SIG_ to 1. The test shows that the sample constrainer is effective at removing sharp glitches.

5.3.3.4 Groove width estimator and 'Open mouth' DC algorithm.

Unlike CD, the groove width of a vinyl disc is not constant. Wider groove width is needed for good tracking when the audio signal is loud. Smaller groove width is needed for reducing the surface area for more content when the audio signal is small. The groove width of one vinyl is therefore not constant. To design robust audio decoding at various groove widths, adaptive groove width estimation is needed. Since groove width is directly related to audio signal amplitude. The maximum amplitude can also be predicted.

The estimated groove width is given by the distance between the left channel DC and right channel DC measured in *clk*. The DC is estimated through a moving average filter of length = 1024 with a zero at 26Hz. The groove width estimation will thus have a bandwidth of around 20Hz. The groove width estimation result will approximately contain only DC caused by groove width and will not be affected by

(L-R) audio signals. The groove width estimation is also proven experimentally to be fast enough to capture changes in groove width.

The groove width estimator outputs the max groove width (**G_{max}**), max allowed jump (**J**), and estimated DC of both channels to the sample constrainer. The sequential update equations are shown below.

$$\mathbf{G}_{max} = \boldsymbol{Estimate\ Groove\ Width} + \mathbf{J} = \boldsymbol{Scan\ range} - \boldsymbol{DC_L} - \boldsymbol{DC_R}$$

$$\mathbf{J} = 0.5 \times \boldsymbol{G_{max}}$$

After the estimation, the groove width estimator will rescale the sample to a value between -32768 to 32767. This is done by (selected_sample/scan_range – 0.5)* 65535. The rescale operation is done by 32 bits floating point calculation therefore it preserves the fidelity of 16 bits audio. The rescaled sample is used in groove tracking. The rescaled sample with **negated** right channel audio is used as audio output (before the audio enhancer).

Since all parameters are adaptive, an initialization function of parameters is required when the system encounters a fault. The initialization function (RESET_CTRL) is set as follows.

$\mathbf{G}_{max} = \mathbf{4000}, \mathbf{J} = \mathbf{2000}, \boldsymbol{DC_L} = \boldsymbol{DC_R} = \boldsymbol{center\ of\ scan}$, **estimated groove width** = 0.

The initialized parameters will be used until the adaptive groove width prediction starts. The adaptive groove width prediction will start when 512 pairs of audio samples are correctly decoded after each RESET_CTRL. The initialization favors the closest pairs of left and right audio edges near the center of scans; therefore, it ensures that the selected pairs of signals will always belong to the same turn of the groove regardless of the groove width. The following diagram shows how $\boldsymbol{DC_L}$ and $\boldsymbol{DC_R}$ varies after each RESET_CTRL. The groove width used in the figure is 5000. Center of scan = 5000. Therefore, actual max groove width **G** = 10000 and **J** = 5000**.**

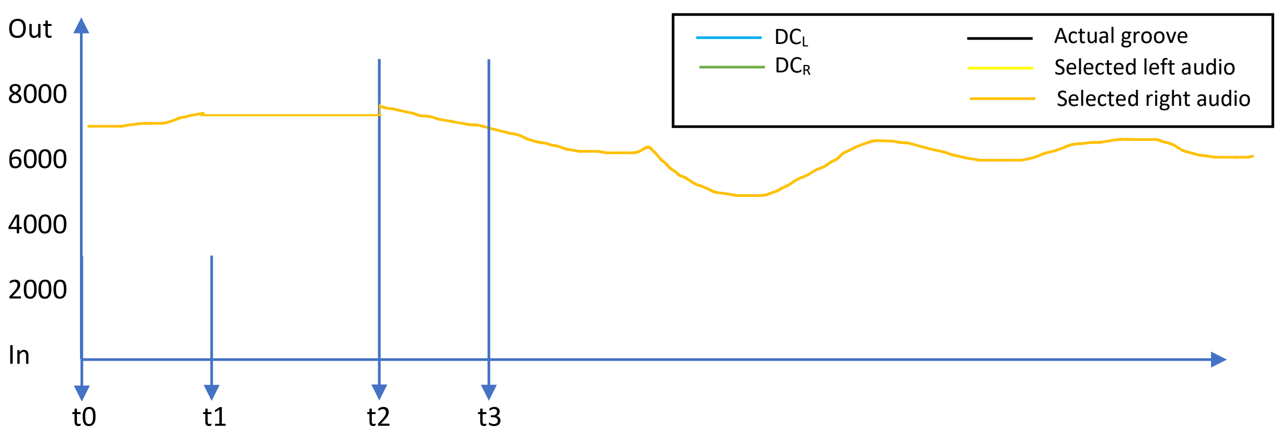


figure 5.3.3.4 a sketch of the internal variables of audio pick-up block after every RESET_CTRL

t0: At this time instance, the audio pick-up is reset. The $\boldsymbol{DC_L}$ and $\boldsymbol{DC_R}$ is initiated to 5000, the last left audio and right audio is also initiated to 5000. The max jump ***J*** is set to 2000 and the max groove width is set to 4000.

From t0 to t1: Within this time, the system will only pick up right channel audio samples. The left channel samples are rejected because their distance to the $\boldsymbol{DC_L}$ is more than ***J*** (2000).

t1: At this time instance, the left channel samples are picked because their distance to the $\boldsymbol{DC_L}$ is less than $\boldsymbol{J}$ (2000).

From t1 to t2: Within this time, the system keeps picking left audio data; however, the distance between the right channel groove and the left channel groove is more than $\mathbf{G_{max}}$ (4000). The right channel samples are rejected. The system keeps extrapolating the previous samples.

t2: at t2, the instantaneous distance between the right channel and the left channel is reduced to **G** (4000). The system starts to pick up the right channel sample from the groove.

From t2 to t3: Within this time, both right and left channel grooves are being picked. 512 pairs of left and right channel samples are picked within this period.

t3: There are more than 512 pairs of right and left channel samples being picked. The adaptive groove width starts to kick in. The gap between $\boldsymbol{DC_L}$ and $\boldsymbol{DC_R}$ is 4000**.** The current $\boldsymbol{J}$ is 2000. Maximum groove width $\mathbf{G_{max}}$ = 4000 + 2000 = 6000. New **J** is thus 6000 x 0.5 = 3000.

After t3: The **G** and **J** will keep updating and converging to the actual values where $\mathbf{G_{max}}$ = 10000 and **J =** 5000**.**

The algorithm will not pick up both channels in 10 seconds if there exists a groove that has no instantaneous groove width < 4000 (40µm) in 10 seconds. Fortunately, experiments and logic show that such grooves do not exist. The **RIAA** standard specified that the groove width can take any value from 30µm to 80µm. This means that if a 10-second groove contains instantaneous groove width from 40µm to 80µm, it will always be optimized by engineers to a groove containing instantaneous groove width from 30µm to 70µm. Such optimization not only causes no degradation to audio but also saves more surface area on the vinyl for more content.

4.3.3.5 Mismatched L-R pairs correction and dust skipping.

All adaptive algorithms could adapt or latch to an incorrect steady state. Therefore, fault detection and correction are essential for the adaptive groove width estimator.

Since the audio sample selector always prefers the closest samples to the $\mathbf{DC_L}$ and $\mathbf{DC_R}$, a fault in groove width and DC estimation could latch the system in fault. The error in DC estimation could occur when there is more than one groove in the scan range. If the system decided to pick up the right channel of the one turn of the groove and the left channel of its inner adjacent groove, the adaptive groove width will then adapt itself to about twice the actual groove width. After that, the system enters a faulty steady state. This is illustrated in figure 4.3.3.4a. We call this situation 'Mismatched L-R pairs'

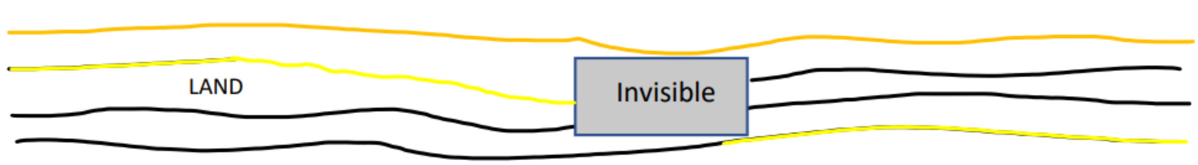


figure 5.3.3.5a. an example of Mismatched L-R pairs

The 'Open Mouth' algorithm indicated in figure 5.3.3.4 ensures that the selected L-R pairs are the closest pair only under the assumption that all grooves are always visible in the SIG; however, a slight reduction in reflectivity of the disc could make edges invisible. After that, the sample selector might pick up the samples from the adjacent track and run into Mismatched L-R pairs.

As illustrated in figure 5.3.3.5a, when Mismatched L-R pairs occur at a visible region, the selected sample pairs of L and R channels will not have consecutive indexes in the SIG_LTR or SIG_RTL buffers. Therefore, this fault can be easily detected by calculating the difference between the indexes of the selected L and the selected R sample.

However, dust in the groove could also cause the absolute difference in the indexes of the selected L and the selected R sample to be none one.

We want a correction to Mismatched L-R pairs, we don't want a 'correction' to dust. The only way to distinguish between these two cases is through accumulating the occurrence of nonconsecutive indexes. The SIG edges of actual groove edges will last longer than that of dust. The system will only start a correction when the time of occurrence (accumulation) of nonconsecutive indexes reaches the threshold.

The rule of the accumulation is set as follows:

1. If both channels have available samples from the sample constrainer, add one to the accumulator if a pair of nonconsecutive indexes are found, and subtract one to the accumulator if a pair of consecutive indexes are found and the accumulator is greater than zero.
2. If either of the channels has no samples from the sample constrainer, the accumulator remains unchanged.

The threshold of the accumulator is determined by dust size. Assuming that the dust is circular, a in groove dust will have a diameter of up to 80μm. The minimum diameter of a turn of the groove is 10.5cm. The minimum circumference is 33cm. If the rotation speed is 33.33RPM, the moving speed of the groove is 11cm/s. It will take 0.08/110 = 727.3μs for the dust to pass the read head. If the sample rate is 26.4kHz, the dust will affect at most 19.2 samples. The final decision of the threshold is set to 50 samples.

The system will force a correction on the right channel when the accumulator is greater than 50.

When a correction is forced, the groove width constrainer will temporarily be off to allow a jump to happen. The system then forces the right channel data to be the closest outer edge next to the selected left sample. The correction will keep forcing the right edge until the sample selector selects a consecutive indexed L-R pair, i.e. until the estimated $\mathbf{DC}_L$ and $\mathbf{DC}_R$ falls to one turn of groove. The constrainer will then be on again and the system should function correctly afterwards.

4.3.3.6 Evaluating and Testing the audio decoding as a block

1. Static groove test

This test is carried out by placing a non-rotating disc under the read head. Since the disc is not turning, we should get a stable groove width estimation, all NO_SIG_ =0, and a quiet audio output at the speaker. The max groove width is printed to the serial terminal. All NO_SIG_ = 1 will be caught.

The system successfully and stably picked up a pair of samples for discs with different groove widths; however, the following problem is detected:

(1). The groove width estimation is correct, but the value fluctuates around the mean value

(2). The audio output is not quite, there is a single-tone high frequency in the audio and some white noise.

(3). Depending on the groove positions, random glitches occur. This is true especially when one of the SIG edges overlaps with either SYNC.

This problem will be solved in Audio Enhancer.

2. Moving groove test

The test is carried out by placing the disc static first and letting the algorithm pick up a pair of stereo samples. The groove tracking system is turned off. Then, gently rotate the disc in either direction, so that the selected groove will appear in any position of the scan.

The audio decoding output shows a slowly varying DC voltage for both channels as the disc rotates. This indicates that the algorithm can pick up the desired groove turn regardless of its relative position in the scan; however when the desired groove turn is close to the edge of the scan range, the noise increase. The test also reveals the same problems discussed in the static groove test.

3. Groove width initialization test

The test is carried out by placing the read head at a random location after the RESET_CTRL is called. The random position includes between-song-spiral, slim groove region, and wide groove region. The DISC is rotated slowly by hand. The algorithm can capture 30um to 60um grooves immediately; however, it may take 1 to 5 seconds to capture an 80um wide groove. This number will get smaller if the disc rotation is faster with groove tracking on.

The slower speed for wide groove width is mainly because the initial value of DC_L and DC_R is at the center of the scan and the max groove width is initialized at 4000 (40um). Initializing the system width bigger max groove width could result in picking mismatched pairs of left and right channels from different turns of the groove. If it happens, the correction will cause a further glitch in the right channel. This is not preferable.

4. Fault detection and correction testing.

To inject mismatched L-R pairs, the $\boldsymbol{DC_R}$ is initialized to the edge of the scan range and the max groove width is initialized to the width of the scan range. A serial print function is called whenever the correction is activated.

After around 100 corrections, the system settles. The DC audio output of the right channel drops from 1V to 0.2V, indicating that the groove width is reduced from about 70um to 30um. The DC audio output of the left channel stays at 0.4V. This indicates that the correction is a success.

5.3.4 Groove tracking control system

The design procedure of the groove tracking control system is very similar to that of the focus tracking control; therefore, the repeated discussion will be omitted throughout this section.

The groove tracking system aims to maintain the selected groove at the center of the scan range. The specification of the groove tracking system is provided in 3.1.3. Groove Tracking and Maximum Trackable Center Deviation

1. Selection of controller type.

The previous groove tracking control system is implemented by a proportional controller (P controller). The driving voltage on the horizontal attenuator is obtained directly from the groove position. The same attenuator as in the focus control is used in the groove tracking system. As discussed in 4.3.2, the spring-mass system requires a PID controller to remove the steady-state error.

2. Choice of the error signal

The error signal of the system should be a low passed version of the groove center position so that the attenuator does not play the (L+R)/2 audio by itself.

One solution is to use the middle of $\mathbf{DC_L}$ and $\mathbf{DC_R}$ from the groove width estimator; however, the bandwidth of $\mathbf{DC_L}$ and $\mathbf{DC_R}$ is only 20Hz and is insufficient for the groove tracking system required by the specifications in 3.1.3.

Instead, the moving average of the rescaled audio outputs $\mathbf{L_n}$ and $\mathbf{R_n}$ from the groove estimator is used. The difference in these moving average audio outputs forms the error signal of the groove tracking controller. The length of this moving average filter can be modified through a variable.

3. The design of the Groove tracking PID controller

(1) Determine the maximum bandwidth of the system

The same approach is used to measure the response of the groove tracking attenuator. Unlike the under-damped focus tracking attenuator, the groove tracking attenuator tends to be overdamped. No zero is needed to cancel out oscillation.

The specification in 3.1.3 suggests a bandwidth of 87.2Hz. This is already at the limit of the mechanical system. The moving average filter for the error signal should therefore be as short as possible to improve reaction speed. The moving average filter should also filter out all frequencies above 4000Hz as these frequencies could generate harsh sound on metallic parts.

The final decision of the moving average filter length is 8.

(2) Modeling and Designing theoretical **kp**, **ki**, and **kd** and mapping to hardware.

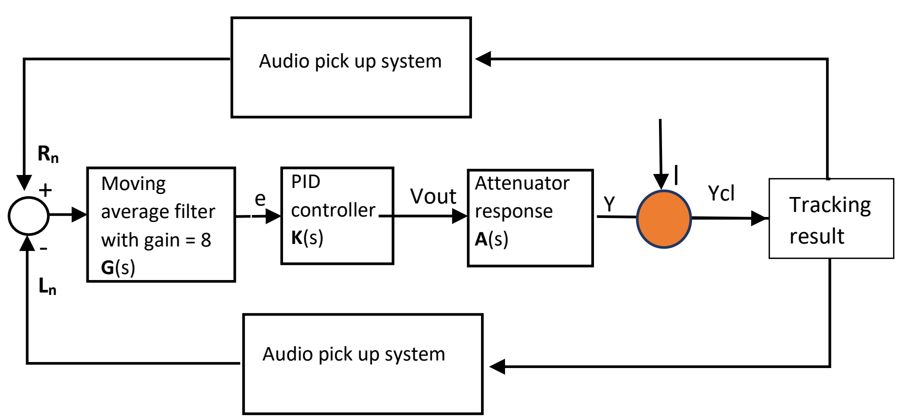


Figure 5.3.4.3. The model of the groove tracking control loop.

After modeling the groove tracking control system, the rest of the design procedures are the same as the focus tracking system. The final decision of digital PID coefficient D_p, D_i, and D_d is $\frac{800}{2^{22}}$, $\frac{18}{2^{22}}$, and $\frac{50}{2^{22}}$.

The output of the PID controller is then fed into two hardware. They are the PWM signals for the groove track attenuator and the analog voltage for the coarse groove track linear stage motor. The PWM converts -256 to 256 into -15V to +15V. The analog voltage converts -256 to 256 into 0 to 5V. The linear stage moves inside with voltage < 2.5V and moves outside with voltage > 2.5V. The moving speed is proportional to voltage.

(3) Dealing with errors and faults.

The design uses the same logic described in the focus control to prevent the integral from overflowing, but this is not adequate.

The coarse tracking linear stage motor should always move in the direction to release the integral. The integral should therefore never overflow, otherwise, the coarse tracking motor must be at its end or have a fault. In such cases, the integral and moving average results should be reset, and a hardware error should be reported.

The groove tracking control system will also be reset under the following circumstance:

a. The audio pickup system raises a **RESET_CTRL** event to find a new groove
b. The system enters **PLAYING** mode from other modes (**FIND_NEXT_SONG, FIND_PREVIOUS_SONG, CHANGE_DISC, LOAD, UNLOAD)**

When there are lots of missed samples from the audio pick-up system, the groove tracking system should keep the coarse tracking motor static to avoid fast-tracking movement caused by the error signal. The enabling and disabling of the coarse groove tracking linear stage is controlled by the following two variables.

Cumulated Correct Count: The counter is incremented by 1 if both L and R channel data is successfully picked and verified by the audio pick-up system. The counter is decreased by 1 if either of the L and R channel data is not successfully picked and verified by the audio pick-up system. The counter value is limited between 0 and 10*Mirror frequency.

Cumulated Missed Count: The counter is incremented by 1 if either L and R channel data is not successfully picked and verified by the audio pick-up system. The counter is cleared to 0 if both L and R channel data is successfully picked and verified by the audio pick-up system.

The coarse groove tracking linear stage motor will only be enabled when **Cumulated Missed Count** < 5000 and **Cumulated Correct Count** > 5000.

5.3.5 Audio Enhancer

After correctly integrating the focus control, audio pickup, and the groove tracking control system together, the player started to play continued audio from records; however, the sound quality is poor.

1. 13.7kHz and white noise present in the audio
2. Random glitches present in the audio
3. 428Hz periodic glitches present in the audio

4. The audio is too bass-heavy.

Since works all run ahead of schedule by 2 weeks. I decide to move on to advance features. The first block to design is the audio enhancer.

5.3.5.1 Dealing with 13.7kHz noise

1. Cause of noise

So far we have assumed that the relative position of a groove edge is proportional to the time difference between the actual time of arrival of the selected sample and _sync_start_frame. This assumption is true only if the scanning speed of the mirror is constant.

The scanning speed of the mirror is near-constant and is fastest when the mirror is in the center of the scan range. The scanning speed is, however, slower at the edge of the scanning range. In another word, the mirror decelerates and then accelerates when it approaches its edges.

Depending on the actual phase and frequency deviation of the mirror oscillation to the driving signal, the deceleration and acceleration may not be symmetric, resulting in unsymmetric speed around SYNCs. A test on the static groove shows that the center of LSYNC pulse is about 300ns slower than the symmetric center of SIG around LSYNC. This lagging is 167ns for RSYNC. The lagging is caused by (speed before SYNCs rise) > (speed after SYNCs fall). The fastest speed region within each scan is shifting slightly towards the end of each scan.

2. Solving the problem from raw data

To compensate for this lagging, 180 is subtracted to lsync_start_frame and 100 is subtracted to rsync_start_frame.

After the correction, the symmetric center of SIG and the center of SYNCs are aligned in static groove cases. This means the audio sample acquired from either scan direction is identical to the specific groove. The 13.7kHz noise is therefore removed for the specific static groove.

3. Solving the problem from processed data

The solution in 2 can only completely remove the 13.7kHz noise for the specific groove under the test. 13.7kHz noise is still audible when a new groove is chosen. Also, the resonance frequency of the scanner changes over time and the lagging is not a constant.

A better method would be an FIR filter. The filter should remove 13.7kHz. Since 13.7kHz is fs/2. If x(n) has a noise = a, then x(n-1) must have a noise = -a and x(n-2) must have a noise = a, assuming that the amplitude of the 13.7kHz is a constant within 3 samples. The FIR filter can now be designed intuitively.

Filter 1: A 2-sample moving average filter is given by x(n)+x(n-1). In this filter, the noise 'a' and '-a' get canceled.

Filter 2: The difference of x(n) and x(n-2) is given by x(n)-x(n-2). In this filter, the noise 'a' is canceled again.

Filter 3: The sum of Filter 1 and Filter 2 will also contain no 13.7kHz noise. It is given by: 2x(n)+x(n-1)-x(n-2)

Filter 4: Designed using a zero in z-transform. The result is given by x(n)+2x(n-1)+x(n-2)

Since traditional stylus vinyl player plays the differentiated groove data. The filter here should ideally have a differential tern to boost higher frequencies. Filter 2 or filter 3 is therefore preferable. Figure 5.3.5.1 plots the frequency response of these four filters.

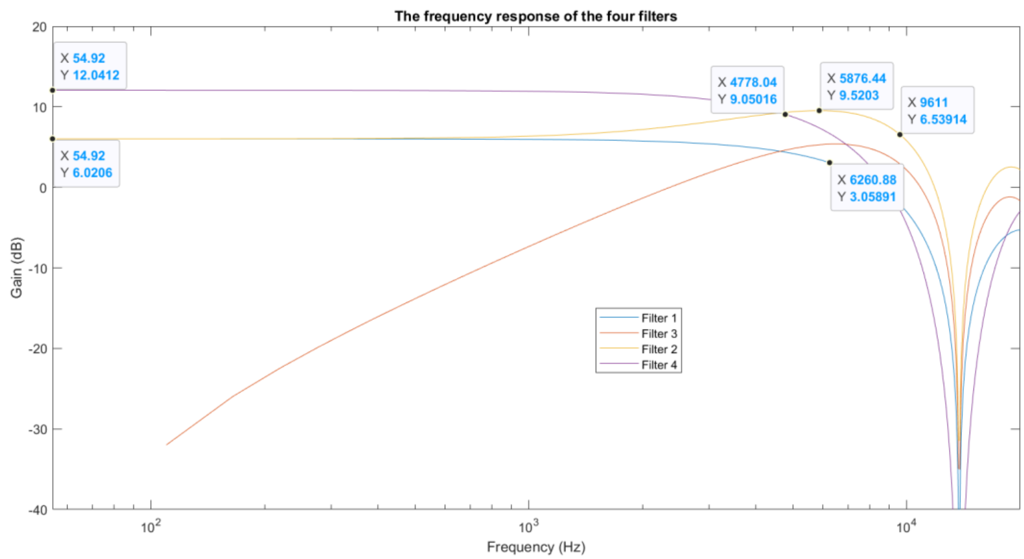


figure 5.3.5.1 frequency response of four filters with -3dB cut-off frequencies.

The -3dB cut-off frequency is the frequency point where the gain is 3dB down from the max gain. Filter 2 has the highest cut-off frequency at 9.6kHz. Filter 2 also boosts all high-frequency components from 1000Hz to 11kHz. This will compensate for the higher bass output due to not using a differentiated groove signal. Also, bass information is preserved indicated by a 6dB gain at DC.

Filters 1 and 4 have a lower cut-off. They do not provide gain for higher frequency to compensate for overwhelmed bass. The only advantage of these two filters is that they are linear phase filters; however, human ears are not sensitive to the phase distortion of audio. (The passive crossovers in loudspeakers are not linear phase filters too.)

Filter 3 corrupts bass. This is not the final stage of the audio output. It is better to preserve as much information as possible at this stage.

In the end, filter 2 is selected. To avoid clipping, all filter 2 coefficients are divided by 4. The filter removes the 13.7KHz noise completely. The sound, however, still lacks mid and high frequencies.

5.3.5.2 Dealing with 438Hz periodic glitches

438 is 13.7k divided by 64. The DMA is filled at exactly every 64 samples. This suggests that the noise is caused by the DMA interrupt generated by the SAI. By default, all interrupts have priority =128. The GPIO interrupt's priority is raised to 16 to allow it to preempt the SAI interrupt as well as other interrupts.

5.3.5.3 Dealing with random glitches and white noise

1. Extrapolation algorithm of the missed sample.

So far, all missed sample from the audio sample pick-up is being estimated by the last available sample. This guarantees that the last available sample transits to the estimated sample smoothly, but it does not guarantee that the estimated sample will transit to the next available sample smoothly. If the last available sample is far away from the next available sample in the time domain, the estimated current sample (equal to the last available sample) could deviate a lot from the next available sample. This results in a glitch.

To avoid the glitch in the 'future', one must connect a straight line from the 'last available sample' to the 'next available sample' (in the future) to estimate all missed samples between them. It is impossible to know the future. The only way to know the 'future' is to buffer the signal for a fixed length and outputs the oldest sample to the audio. The current sample then becomes the 'future'. All older elements inside the buffer can be modified by their 'futures'. The length of the buffer should be determined first.

We define the Consecutive_Missed counter, we add one to the counter whenever we have an estimated sample. We **record** the value of the counter and reset it to zero whenever we have an available sample picked from the groove. The following diagram shows the value of the 1090 **record** of the Consecutive_Missed counter

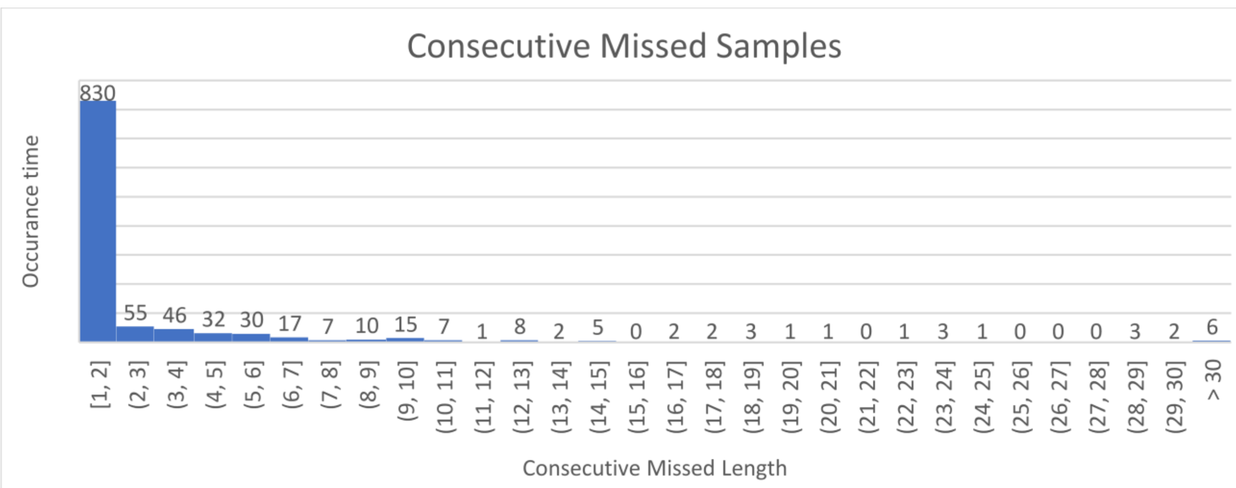


Figure 5.3.5.3a Consecutive missed sample during the 20s playing 33RPM vinyl.

2627 out of 528000 samples are missed, the missing sample rate is 0.498%.

In about 76.1% of the missed sample cases, we have less than 2 consecutive missed samples.

In about 81.2% of the missed sample cases, we have less than 3 consecutive missed samples.

In about 85.4% of the missed sample cases, we have less than 4 consecutive missed samples.

Numbers of consecutive missed sample	The probability, given that missed sample occurs
<2	0.761
<3	0.812
<4	0.854
<5	0.883

<6	0.911
<7	0.927
<8	0.933

Table 5.3.5.3a Cumulated probability of consecutive missed sample.

The result in table 5.3.5.3a suggests that we would deglitch 93.3% of the missed sample cases if we have a buffer length of 8. If we have consecutive missed samples >= 8, we can still extrapolate the 'next available sample' with the last 7 estimated samples to create a smoother transition. The final decision on the buffer length is 8. Having a larger buffer (such as 16) is possible and configurable, but the marginal performance is little.

The algorithm of the extrapolator is summarized below, the algorithm is implemented for both L and R channels. The algorithm can only modify the estimated sample, it can not modify the picked sample.

Note that the phrase 'roll the **Denoise_RC**' means add or subtract one to **Denoise_RC**. When the operation gets **Denoise_RC** over 7, we reset **Denoise_RC** to 0. When the operation gets **Denoise_RC** below 0, we set **Denoise_RC** to 7.

a. When a new audio sample arrives, it checks whether there is a new **record** of the Consecutive_Missed counter. If not, skips to 'e'. If yes, goes to 'b'.
b. If the **record** is bigger than 7, we set **extrapolate_length** to 7. Otherwise, we set **extrapolate_length** to **record**.
c. Take the buffer, and roll the **Denoise_RC** pointer back by (**extrapolate_length+1**) to acquire the 'last available sample'. Using the newly arrived sample as the 'next available sample', calculate the difference between them. The difference is then divided by (**extrapolate_length** +1) to yield the **slope**.
d. Roll the **Denoise_RC** forward step by step until it is equal to its value before 'c'. Within each **step**, the extrapolated sample value is equal to 'last available sample' + (number of **steps** taken * **slope**). Store the extrapolated sample value to the position pointed by the **Denoise_RC** in each step.
e. Send the sample pointed by **Denoise_RC** before 'c' to the audio output.
f. Store the newly arrived sample to the position in the buffer pointed by **Denoise_RC** before 'c'**.**
g. Roll the **Denoise_RC** forward by 1.

2. Adaptive Frequency Lock Loop for quieter RSYNC and LSYNC

The RSYNC and LSYNC signals are generated from comparators. The analog signal before the comparator will have noise caused by interference from other signals. The noise before the comparators could lead to shifting in RSYNC and LSYNC signals. This shift causes glitches in audio since all audio signals are decoded from the SYNCs.

During the detection of the SYNCs, the overlapping of interrupts could also shift the SYNCs. According to the timing analysis section, the SIG interrupts need 11(latency)+36(process) clock cycles to complete. If a SYNC overlapped with a SIG. Its detection time could be delayed up to 47 clock cycles. Such delay also causes glitches.

Since the mirror is driven by an onboard signal, it looks straightforward to design a phase lock system to lock the LSYNC and RSYNC to the onboard driving signal. However, this method is not reliable. Because the phase shift depends on the deviation of the driving signal frequency to the actual mirror oscillation frequency. The mirror oscillation frequency depends on temperature. The phase shift also depends on the driving force.

This means that the RSYNC and LSYNC are needed to be captured from hardware, but they should not be used directly for audio decoding.

Since the scanner mirror utilizes oscillation, its frequency (period) and phase shift can not change within a small time interval. Utilizing this property, we can design an adaptive frequency lock loop to generate quieter SYNCs.

For the n^{th} scan, Let $\mathbf{T_{LTR}}(n)$ be the time (clk) taken for the mirror to scan from its left end to its right end. Let $\mathbf{T_{RTL}}(n)$ be the time taken for the mirror to scan from its right end to its left end.

If the mirror frequency and phase do not change within k back-and-forward scans, we have:

$\mathbf{T_{LTR}}(n) = \mathbf{T_{LTR}}(n-1) = \mathbf{T_{LTR}}(n-k) = \mathbf{T_{LTR}}$ and $\mathbf{T_{RTL}}(n) = \mathbf{T_{RTL}}(n-1) = \mathbf{T_{RTL}}(n-k) = \mathbf{T_{RTL}}$, where k is any real integer.

We never know the exact value of $\mathbf{T_{LTR}}$ and $\mathbf{T_{RTL}}$, but we can guess it and adaptively adjust its value smoothly to approach the actual value of $\mathbf{T_{LTR}}$ and $\mathbf{T_{RTL}}$. according to the measured SYNCs. 'Smoothly adjust' means we can only add 1 or subtract 1 to our estimation $\hat{\boldsymbol{T}}_{\mathbf{LTR}}$ and $\hat{\boldsymbol{T}}_{\mathbf{RTL}}$ for each back-and-forward scan. 'According to the measured SYNCs' means using the measured intervals $\dot{\boldsymbol{T}}_{\mathbf{LTR}}$ and $\dot{\boldsymbol{T}}_{\mathbf{RTL}}$ from the pulse centers of hardware SYNCs to determine whether we add 1 or subtract 1 to $\hat{\boldsymbol{T}}_{\mathbf{LTR}}$ and $\hat{\boldsymbol{T}}_{\mathbf{RTL}}$. The following figure shows the detail of the algorithm for $\hat{\boldsymbol{T}}_{\mathbf{LTR}}$ at DIR_LTR = 0. The same algorithm is used for $\hat{\boldsymbol{T}}_{\mathbf{RTL}}$ at DIR_LTR = 2. The content in the following figure can only run **a single time** within a single back-and-forward scan.

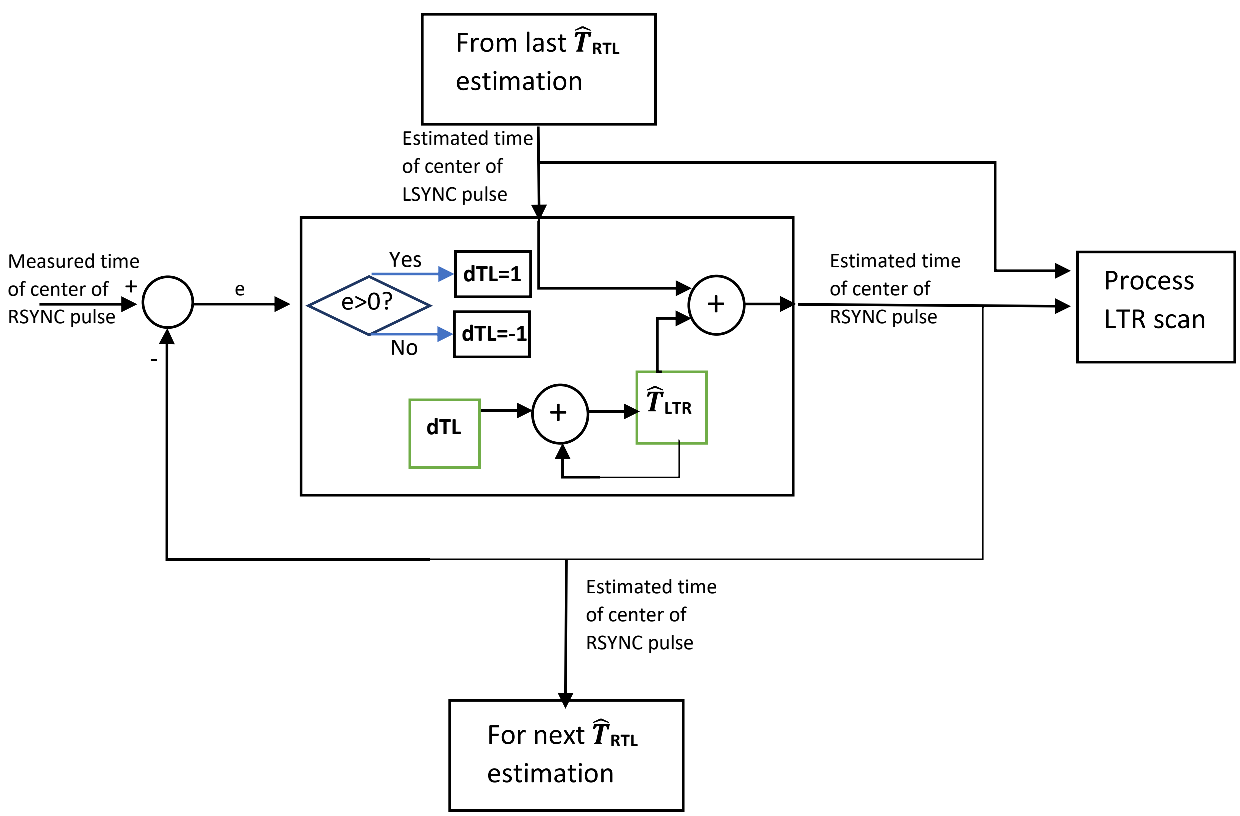


Figure 5.3.5.3b. the flow chart of the algorithm for estimating $\hat{\boldsymbol{T}}_{\mathbf{LTR}}$ at DIR_LTR = 0

The adaptive algorithm needs fault detection and initialization. $\hat{\boldsymbol{T}}_{\mathbf{LTR}}$ and $\hat{\boldsymbol{T}}_{\mathbf{RTL}}$ are initialized to $\dot{\boldsymbol{T}}_{\mathbf{LTR}}$ and $\dot{\boldsymbol{T}}_{\mathbf{RTL}}$ at the start of the program and at $|e| > 5000$.

The following figures compare the Left channel audio output with and without the Adaptive Frequency Lock Loop playing sine frequency sweep.

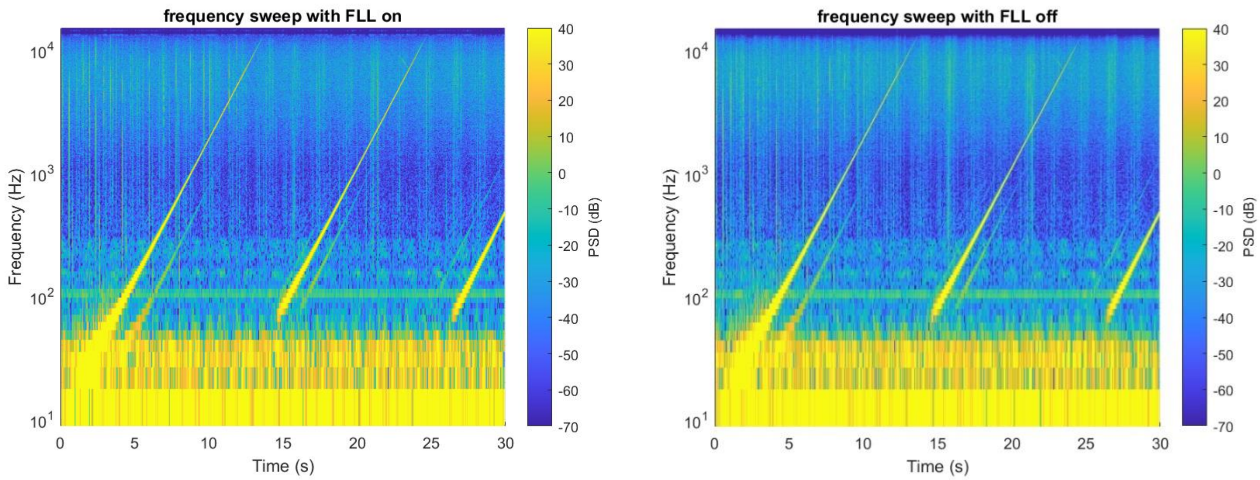


Figure 5.3.5.3c. Comparison of the time-varying frequency spectrum of FLL disabled and FLL enabled for the left channel with recommended EQ settings.

Glitches will appear as vertical strips cutting the frequency-time spectrum. The brighter the strips, the bigger the glitches. The frequency-time spectrum has noticeable brighter strips at T=15s, 22s, and 27s when FLL is off. The noise within 2000 to 11000Hz is stronger when FLL is off. This indicates that the FLL successfully reduces the amplitude of peaks and reduces the power of high-frequency noise. The listening test also suggests the same result.

Further noise reduction will require modification of hardware. Modifying hardware is risky since it could lead to other unexpected issues.

The remaining noise in the system is mainly white. White noise is unpredictable, one way to remove white noise is through predictive AR models of music and noise pattern using AI. I have tried all noise removal in **Audacity**, and all advanced non-real-time noise removal in **Audacity** could not effectively remove the remaining noise without causing severe degradation to music quality. The real-time noise removal on a 600MHz single-core MCU will always be worse than non-real-time noise removal on a powerful computer. This suggests that investigating time on those algorithms might not be worth it.

5.3.5.4 Investigation of channel separation and linearity

The following figure is a magnified version of Figure 4.3.5.3c FLL off. Labels are omitted to save space. This time the right channel data is shown instead of the left channel. I haven't done the FLL when I am analyzing this part, therefore the test is carried out without FLL.

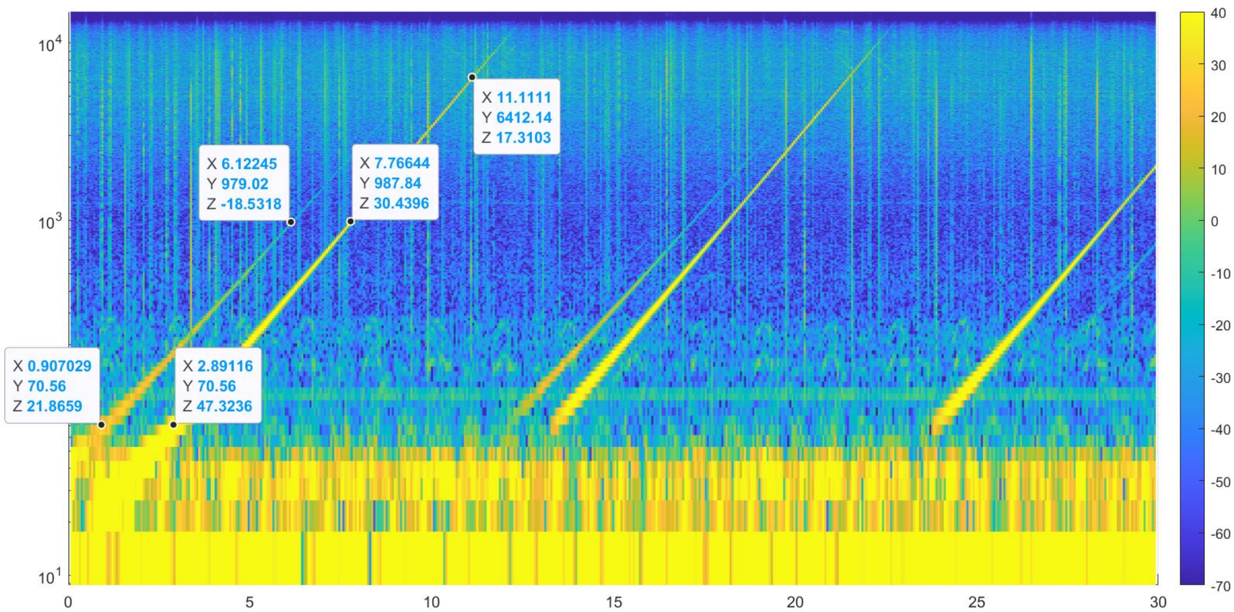


Figure 5.3.5.4. the frequency sweep result of the right channel at FLL off with recommended EQ settings.

The track in Cardas Burn is a 30Hz to 20kHz frequency sweep track. The track is meant to be played at 45rpm. The track consists of an asynchronous frequency sweep for both left and right channels. i.e., the instantaneous frequency of a channel is always ‘k’ times higher than the other channel within each sweep, where 0.5 < k < 2.

The channel separation is carried out at 33rpm because I want to inject lower frequency signals which are more likely to have an issue with the groove tracking system.

The groove tracking system is stable throughout the test, the noise between 20Hz to 40Hz shown in the figure cannot be heard in the audio output with good earphones. That noise is not in the audio output. They are caused by the spectrum leakage of DCs and subsonic noise during windowing.

A normal turntable has a channel separation of around 30dB. For this laser turntable, channel separation is 25.5dB at 100Hz, 48.97dB at 1000Hz, and higher for higher frequencies. The low channel separation in the bass is partially due to the control system. The same test carried out with a 5Hz bandwidth groove tracking setting shows only 3dB more separation for 100Hz. The low separation could be due to the cutting of the testing vinyl.

Harmonic distortion caused by non-linearity is very low. An n^{th} order distortion will appear as another line parallel to the third sweep signal where the same signals are applied to both channels. Such lines are not visible and are at least 90dB smaller than the sweep signal.

5.3.5.5 Improving frequency response of the final audio output.

The preprocessor of the SGTL5000 audio codec is enabled to correct the frequency response. The frequency response is smooth but lacks treble and mid-range. The coarse correction is done by two 12db/Oct filters modifying bass below 200Hz and treble above 2000Hz. The frequency response is first coarse tuned using a listening test so that the treble is visible on a time-frequency spectrum diagram. The coarse tune yields -6dB for bass below 200Hz and +6dB for treble over 2000Hz.

The Cardas Burn is rotated at 45rpm with the EQ set to the coarse tune. The frequency response is plotted below.

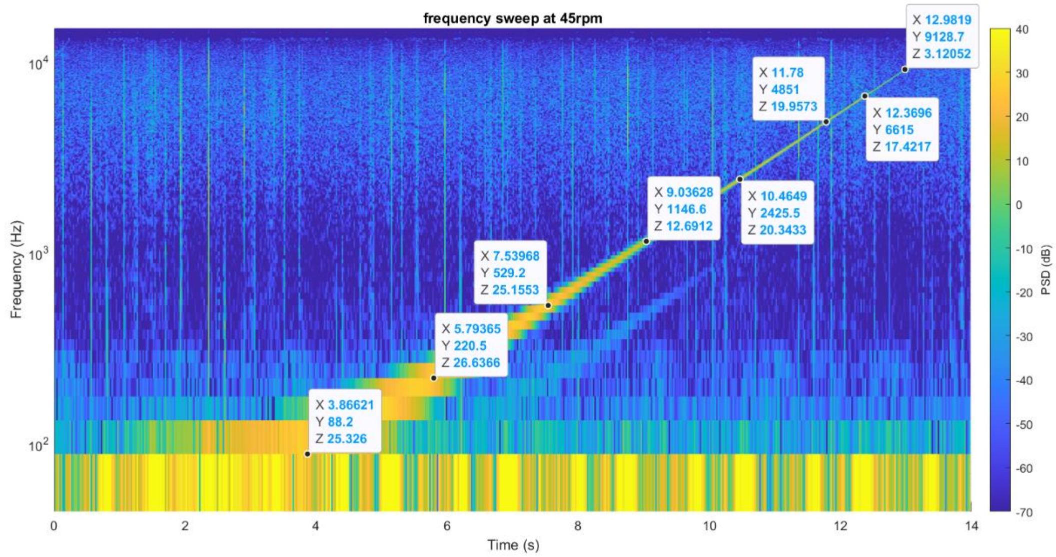


Figure 5.3.5.5. The frequency-time spectrum of the OVDP under coarse tuned EQ with all audio enhancers on.

The rate of change of frequency is not constant throughout the sweep. The rate is slower at lower frequencies and faster at higher frequencies. This results in a reduction in measured amplitude at higher frequencies especially a bigger window is used. A small window, however, does not provide enough resolution at bass. Therefore, a window size of 1000 is used for the analysis.

The sudden drop in gain at 9.2kHz is caused by to fast rate of change in frequency. A window size of 100 (3.6ms) is used to reevaluate its gain. The new gain is reevaluated to be 10dB.

The frequency response and a suggestion for the fine-tune are summarized in the following table.

frequency	Gain after coarse tune	The gain in fine-tune	Final output
100Hz	25dB	-2dB	23dB
200Hz	27dB	-2dB	25dB
500Hz	25dB	0dB	25dB
1100Hz	13dB	10dB	23dB
2500Hz	20dB	3dB	23dB
5000Hz	20dB	3dB	23dB
6600Hz	17dB	3dB	20dB
9200Hz	3dB	3dB	13dB

Table 5.3.5.5. The frequency-response of key frequencies and a suggestion of fine-tune.

The suggested fine-tune can be achieved by:

1. Set the gain of bass in SGTL5000 to -8dB
2. Set the gain of treble in SGTL5000 to +9dB
3. Add a band pass filter at 1kHz to boost 1kHz by 10dB.

The fine-tune should improve the +-3dB bandwidth of the OVDP to at least 40Hz to 8kHz.

The SGTL5000 provides 5 band equalizer which is used for fine-tuning. The third band is used to boost 1kHz. Other bands are used for bass and treble.

The frequencies above 10kHz cannot be boosted further due to noise, they should be enhanced by a smaller laser dot used for the scanner and a scanning mirror with a higher oscillation frequency.

5.3.5 "CD player" functions

The next advanced function to develop is a group of functions that can be found on a CD player

1. Play and pause
   This function implementation is straightforward. When pause is pressed, the system freezes the groove tracking control system and audio pick up. The freeze is implemented by stopping the coarse groove tracking linear stage while keeping all other variables in the ProcessAudioControl unchanged. When play is pressed, the system unfreezes itself and continues playing.
2. Raw back and forward.
   The function allows the user to move back and forward to skip 5-10s audio content. If the buttons are held, the function will keep skipping meanwhile providing the user with audio after each skipping.
   The function is implemented through pause and play. When the button is pressed and held, the system pauses the playing and moves the coarse tracking stage backward or forward for 200ms, then start the playing again for 500ms. The process repeats until the button is released. To improve speed, the max groove width is initialized at 6000 after each skipping.
3. Skip back and forward
   This function allows the user to skip a complete track. The function is implemented as follows:
   a). When the button is pressed, the system starts to move the coarse track motor backward or forward. The algorithm then needs to accumulate 13700 nonempty scans to confirm that the read head is within a song. If so, goes to b, otherwise, stay in a.
   b). The read head is within a song, the next step is to monitor the scans to capture the moment when the read head is out of this current song. The algorithm needs to accumulate 27 empty scans with SIG = 1 to confirm that the read head is out of the current song. After that, the coarse track motor stops and the system starts to play.

   The track detection algorithm is only possible when the system is in open-loop mode. It is very challenging to reliably detect the start and end of a track when the system is in close loop mode. Therefore, the system will not display the track number to the user.

4. Load disc

   This function can only be called when the read head is outside of the disc (after calling Unload disc or reset). Otherwise, it is the same as 'skip back'. The function moves the read head to the start of the first track. The function works for all disc sizes.

   a). When the button is pressed, the system starts to move the coarse track motor forward. The algorithm then needs to accumulate 13700 nonempty scans to confirm that the read head is within a song (must be the first track in this case). If so, goes to b, otherwise stays in a.

b). The read head is within the first song. The next step is to move the coarse track motor backward while monitoring the scans to capture the moment when the read head is out of the first song. The algorithm needs to accumulate 27 empty scans with SIG = 1 to confirm that the read head is at the start of the first song. After that, the coarse track motor stops and the system starts to play.

5. Unload the disc
   The function will move the read head to exactly the edge of the current disc. This allows the user to play the next disc of the same size with ease.
   The function is implemented by moving the read head outward until it reaches a region with an empty scan and SIG = 0.

6. Reset the read head to the outmost position
   This function simply moves the coarse tracking linear stage to its left.

The CD functions work correctly given that the focus control is functioning.

5.3.6 Touch screen and groove visualizer

The user must be able to control the optical vinyl disc player. The control buttons are being implemented by a touch screen. This provides flexibility in a future system update. The touch screen uses an XPT2046 to send touch data to the master and uses an ILI9341 to receive display data from the master. All data is transmitted with the SPI bus. Adafruit ILI9341 library and XPT2046 library are used to ease the development.

The following figure shows the GUI designed for the OVPD

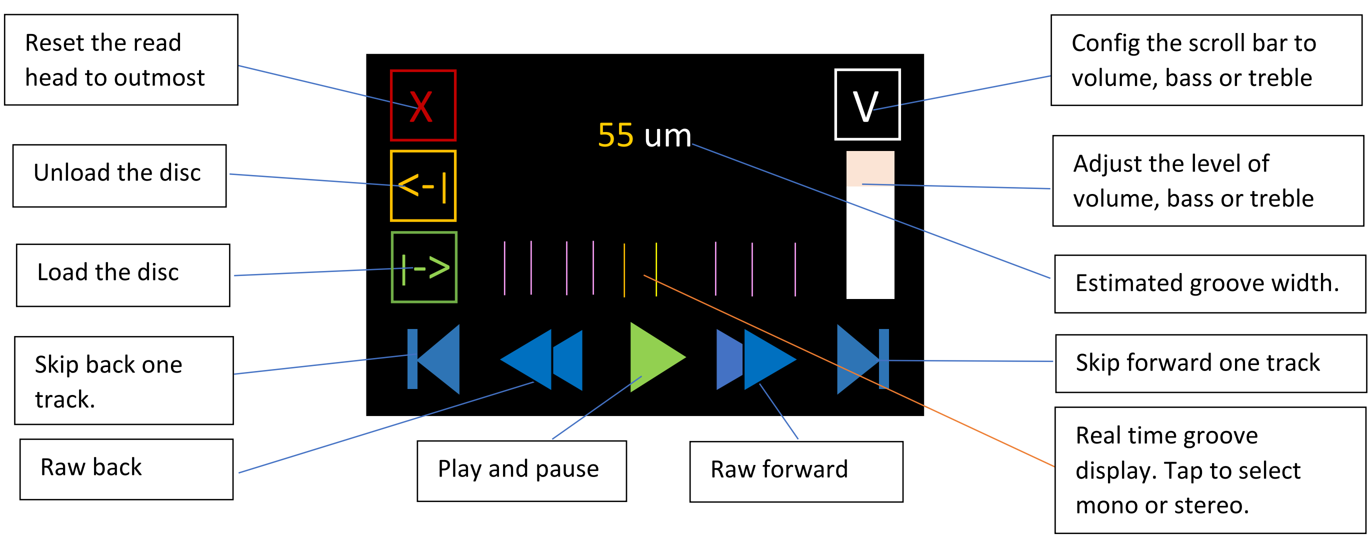


Figure 5.3.6. the GUI for the optical vinyl player

The display update rate is limited to 26Hz. The display update task is the idle task in the system. It will only be activated when the system finishes all other tasks within one audio sample. It can be preempted by any interrupts within the system.

The display task is written in the main loop of the program. The display library is blocking, which means it will block the main loop when it is updating the screen. This means it will block anything else in the loop when it requires more time. Due to this property, the audio processing code must be placed and called by another timer interrupt to avoid discontinuity in audio. The timer interrupt will cause a glitch if a SIG edge occurs at the same time. The maximum value of the glitch is 210 CLK (the latency of initializing the timer in SYNC fall interrupts). The glitch will not occur if the select groove edges are in the center of the scan range.

To solve this, we can simply use two MCU. One of them drives the control system and updates the display. The other one decodes the audio-only. The two MCUs can communicate with UART. I will provide another code with no display and with audio processing in the main loop for the submission. The MCU programmed with the no-display code can be controlled via Serial UART.

Another solution is to use mechanical buttons for control and use LEDs to illustrate states.

The two solutions will not be implemented due to time constrain, and the improvement they provided is negligible compared to other developed blocks.

5.3.7. Play cracked mono shellac discs

The next advanced feature to develop is adding the support for playing shellac disc. Shellac discs have wider groove width which exceeds the scan range of the scanner. To make the shellac disc playable, a single-edge playing mode should be developed. In single-edge playing mode, the user can select either the inner edge or outer edge of a groove. The audio pick-up will only pick up samples from one edge instead of two. The control system will maintain one edge at the center of the scan.

The modification to the sample selector is straightforward. The sample selector only outputs 'left audio sample' for both the left and right channel and groove tracking system when the user selects the inner edge. The same logic applies to outer edge playback.

After such modification, the max jump in the groove width estimator must be modified to make sure that the sample constrainer does not constrain all the audio. The max jump is fixed to ¼ of the scan range. The anti-mismatched-left-right-channel block is being disabled.

This function can be assessed by pressing the groove visualizer on the center of the screen. The user can identify the mode by viewing the groove visualizer. A yellow line indicates a selected inner groove edge. An orange line indicates a selected outer groove edge.

After developing this part, I place an unglued cracked shellac disc on the turntable to play. Due to the effect of the groove position (DC) estimation and the property of non-contact, the player can play a cracked disc as long as the pieces of the disc are placed approximately at their original position. The algorithm can jump to the closest groove at the crack. As long as the closest groove is the same turn of groove, the algorithm can play the audio continuously.

The waveform of the OVDP playing the inner edge of the cracked E11454 disc is shown below. The disc is broken into two pieces. Bigger one 'Piece 1'. Smaller one 'Piece 2'.

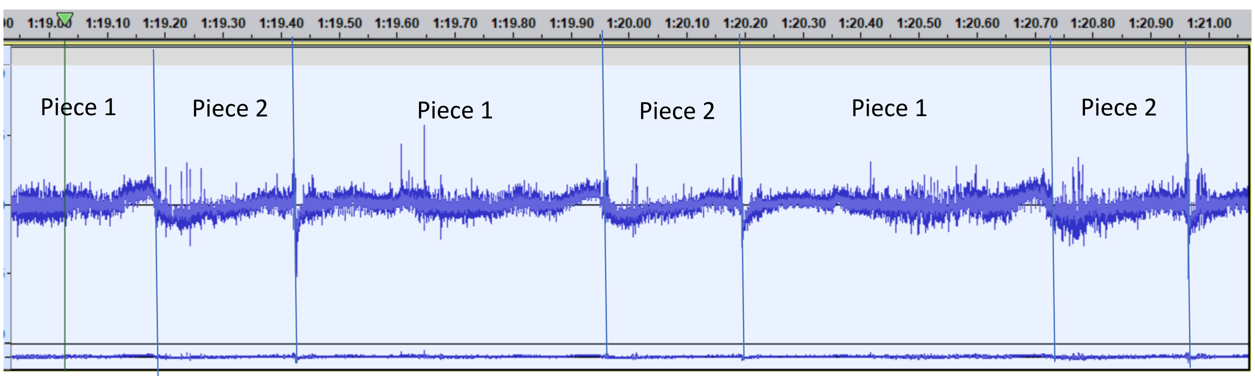


Figure 5.3.7 Waveform of playing a cracked shellac disc. The turntable does not support 78rpm. The audio is recorded at 45rpm and is speeded up to 78rpm in software.

The ability to play a cracked disc without gluing the disc is very valuable for archeology. Also, the cracks on OVDP sound smoother than that on my Columbia 118A gramophone. The high frequency is also richer.

One problem with playing shellac disc is focus issues. Old and worn Shellac discs do not reflect enough light to allow focus tracking to work stably. Also, some old and worn shellac discs could generate too much background noise. In such a case, the OVDP can only be used to recover verbal content.

6. Testing and Evaluation

The testing of all single blocks is discussed in section 5. The whole system will not function properly if one of the blocks failed. They all pass their basic function test. The following table summarized some minor problems remaining.

Block name	**Achieved**	**Remaining issues**
Focus tracking	Keep the scanner in focus	Limited trackable range
Groove tracking	Keep the selected groove at the center	None
Audio pick up	Scan and process audio	Glitches, mirror noise, and white noise
Audio enhancer	Improve sound quality	Some white noise remains, low bandwidth
‘CD player’ functions	Play vinyl discs like playing CDs	No track number information
Display	User-friendly interface	Could cause noise in the audio

The minor issue remaining does not cause problems in the complete system testing. Some of the solutions to the minor issues are discussed in section 5. They will also be summarized in section 7. This part will focus on the evaluation of the complete system.

6.1 Timing analysis, CPU utilization, and maximum sample rate

This section aims to analyze whether the CPU can complete the tasks to meet the timing requirement under the worse case.

The worst case is defined as the combination of the following cases:

a). There are five grooves within the scan range. 10 edges.

b). The adaptive groove width estimator is active.

c). The extrapolator needs to extrapolate 7 samples.

d). The system needs to switch to a new block of DMA.

The tests are carried out by another modified program for testing only. The program generates the groove data and conditions by itself. The following table shows the testing result of all functions.

Functions	CPU Cycles	Latency	Priority
Capturing the rise of any SYNC	4	11	16
Capturing the fall of any SYNC and initialize **Timer1**	212	11	16
Capturing the rise or fall of SIG	36	11	16
Process audio (10 edges) and groove control	1621	13	128
Display (idle)	Remaining	None	Lowest

Table 6.1a Timing of functions and interrupts

The CPU utilization is defined as the percentage of time of CPU processing non-idle tasks. The total processing time of capturing and processing 10 edges within one LTR or RTL scan is:

212 + 11 + 4 + 11 + (36 + 11) * 10 + 1621 + 13 = 2342 cycles. The total CPU cycles within one LTR or RTL is 300M/(Mirror Frequency).

For the current system with Mirror Frequency = 13.7k, the total CPU cycles = 21898 cycles.

The CPU utilization rate is 2197/21898 = 10.7%.

This suggests that the system meets the timing requirement. The CPU's processing speed will not be a bottleneck for Mirror Frequency up to 30kHz. The bottleneck; however, is quantization noise. The theoretical bandwidth of the software, therefore, exceeds beyond 20kHz,

6.2 Glitch-less region

After the timing analysis, we can determine the available position for the selected pairs of edges such that its time measurement will not be delayed by other interrupts. Two SIG edges must be at least 78ns apart. The delay in the measurement of SYNCs will not cause a glitch in the audio due to the FLL system in the audio enhancer. The green region shows the glitch-less region of SIG edges. The figure is not drawn to scale.

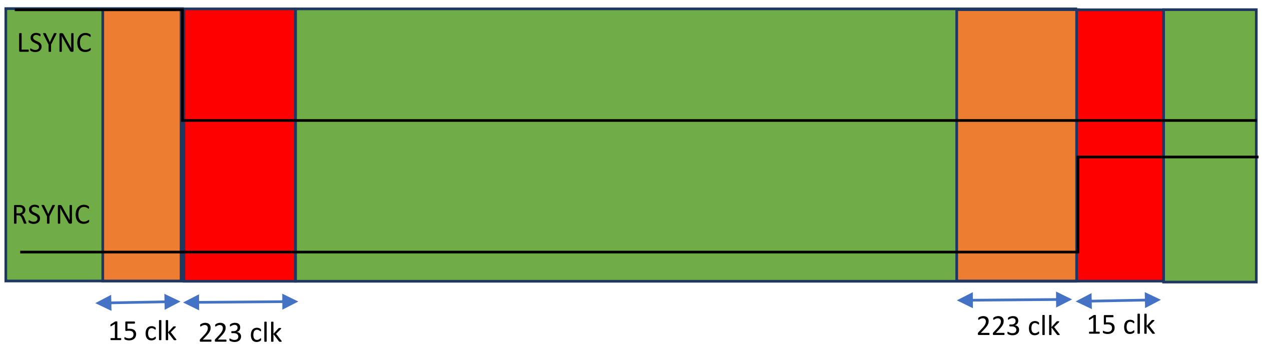


Figure 6.2 The glitch-less region of SIG edges.

Slimmer LSYNC and RSYNC pulse will lead to a bigger glitch-less region in the middle.

In the no-display version, timer1 is not needed. The time for **Capturing the fall of any SYNC is** reduced to 30 clks.

6.3 Headroom of Groove Tracking

This test aims to evaluate the OVDP for DJ applications. A maxi-groove 12-inch disc is used to carry out the test.

The OVDP is set to playing mode. The disc is rotated to up to 100RPM. The system did not lose the groove for at least 10 seconds at 100RPM rotation speed.

I then create some sudden stops, suddenly accelerate backward… to mimic a DJ. The OVDP does not lose the groove if the disc center hole is tight. However, a disc with a slightly bigger hole can cause trouble.

When I test a disc with a worn and loosed center hole, I notice that if my hand causes a sudden left-right shift to the disc during scratching, the groove track may lose the groove for a second and start to play (another groove) afterward. This indicates that the tracking system's bandwidth is still not adequate in terms of heavy-duty DJ applications. Higher bandwidth groove tracking systems do require a smaller weight of the read head and do degrade the bass quality and channel separation.

In terms of normal music listening, the groove tracking system can stably track Discs with a center deviation < 1.5mm up to 100RPM. Discs with center deviation > 1.5mm turned out to be rare.

The groove tracking system can not track the lead spiral at the very end of records. The spiral's moving speed is higher than the maximum speed of the linear stage motor. This is not a severe issue because there is no audio content after this part of the groove.

6.4 Testing results on various disc

I have tested the final version of the OVDP with fifteen 12-inch records. The table shows 10 of them. Only side A of each disc is tested due to time constrain. The result is summarized below.

Title	Number of tracks	Color	Speed (RPM)	Brand	Weight	Testing results and description
Freestyle (EPO)	5	Black	33	MIDI.inc	140g	All pass. Except for the quiet end at track 4 Issue: invisible edge due to low reflection
Variety (2021) Mariya Takeuchi	5	Black	33	Warner music	220g	All pass, but focus control reaches the limit
Woman in red OST Stevie Wonder	5	Black	33	Motown	140g	All pass. Disc has a big center hole and has very fast groove width variation.
Friday Magic Meiko Nakahara	5	Black	33	TOSHIBA EMI	140g	All pass.
Vitamin E.P.O (EPO)	5	black	33	AVG.inc	140g	All pass.
Best Selection 10+1 Meiko Nakahara	6	black	33	TOSHIBA EMI	140g	All pass.
Unknown (Fat boy slim)	1 (Maxi groove)	black	33	unknown	140g	All pass with clipping. Audio signal runs out of scan range due to high amplitude
One last kiss (Utada Hikaru)	1 (Maxi groove)	Clear blue	33	Sony	180g	All pass. Need to adjust the threshold of the SIG comparator.
Best Album (B) (Hatsumi Shibata)	7 (Slim groove)	Black	33	Columbia	140g	Poor SNR. Failed at end of some tracks. Issue: invisible edge due to low reflection.
Debussy Nocturnes /Iberia	3	Black	33	Decca	180g	All pass, but poor SNR when music is quiet.
Abbey Road (Beatles)	6	Black	33	EMI	220g	Only bass is audible.

Table 6.4a The testing result on ten 12-inch vinyl

80% of the testing vinyl discs pass the test. It is not surprising to see failures in playing some vinyl discs, because the vinyl disc is never designed to be played with optics. The cause of failures should still be analyzed.

Some conclusions can be made from the test samples:

1. A vinyl with <7 tracks on each side can be played continuously; the audio quality is good.
2. A vinyl with >6 tracks could run into tracking error due to the invisibility of some dense groove region; the audio quality might be poor.
3. All colored vinyl may require a different threshold for the SIG comparator.
4. All maxi groove vinyl may be subject to audio clipping.

Nealy all originally released album contains 5 to 6 songs per side. Therefore, having a system that favors such vinyl is reasonable. The solution to Poor SNR and 'invisible edge due to low reflection' will be discussed in section 7.

7 inches vinyl generally does not cause failure in groove tracking. They generally produce better sound quality.

The goal of this project is to build a stereo optical vinyl disc player that can play at least 20 seconds without losing the groove. The OVDP so far can play one side (20 minutes) of 12-inch records without losing the groove. The output audio is stereo. Both focus tracking and groove tracking system have been improved by PID controller and moving average filters. The audio decoding module is improved by sample selecting mechanisms. The new audio decoder on MCU is also capable of handling mirror frequency up to at least 30kHz. The software is therefore capable of an audio bandwidth up to at least 20kHz. The frequency response, channel separation, and SNR are also improved by the audio enhancer. Advance features such as CD functions and touch screens are successfully implemented. Since all targets are achieved, I consider the project to be a success.

7. Future works

Though the OVDP is working properly on many discs, its performance is still worse than a stylus player. I suggest further improving the prototype in the following aspects.

1. **Disc compatibility improvement**
   The incompatibility of the slim groove disc is due to invisible edges in the low reflective region. When the tracking system fails, the read head is always at a darker region of the disc. The darker region of the disc is mainly due to the small groove width and close groove distance. Such a region lacks the flat 'land' between grooves.
   The solution would be to use a brighter laser beam for the scanner. When increasing the power of the laser, the heat generated by the laser dot should also be evaluated carefully. The heat on the surface of the record should not exceed 100 degrees Celsius (the temperature of a playing stylus tip is 200 degrees). A protection algorithm should also be implemented in software. The software should reduce the laser power when the disc is not rotating or the SYNCs are missing to avoid overheating the disc.
   A focusing system with a bigger tracking range is also needed. Running out of the focus control range could lead to noise and groove tracking failure; however, reducing the accuracy of the focus control system will only reduce high-frequency response if the error is less than 100um. The design of the focus control system can sacrifice some accuracy to increase tracking range.

2. **Audio quality improvement**
   A dedicated audio decoder MCU or FPGA should be introduced to improve audio quality. The dedicated audio decoder will have the same algorithm as the 'Audio pick up' block in the current

system. If FPGA is used, the clock frequency should be multiplied to at least 600MHz. The audio decoder only takes one control: Play or pause. The audio decoder should have built-in volume control. Advanced signal processing technics may be applied to the audio decoder to further improve sound quality.
Another pair of lasers can be used to pick up pure analog waveform through reflection. The lasers' position can be driven by the error of the current groove tracking system. The lasers' separation can be driven by the estimated groove width.

3. **More functions**
The position of the coarse tracking motor should be fully digitalized. The moving speed of the coarse tracking motor should be increased to allow a pre-scan throughout the entire disc in less than 20 seconds. When a new disc is loaded, the player can analyze the entire disc to store the position of each track. The track number information will therefore be available during playing.

Reference

[1] BLAŽ REMIC, VINYL RESURGENCE-- HOW MATERIALITY AFFECTS ECONOMIC OUTCOMES

[2] History of Vinyl Records | Victrola – Victrola.com 21st January 2022

[3] *Laetare Rose,* 5 Reasons Why Vinyl Records Are Alive and Rising in Popularity. Jun 01, 2020 08:06 PM EDT

[4] Final Year Project Report 2007, Optical Vinyl Disc Reader, Mr. Matthew, Wallace.

[5] ELP Laser Turntable | (elpj.com)

[6] Final Year Project Report 2004, Optical Vinyl Disc Player, by Mr. A. Welton.

[7] Final Year Project Report 2006, Optical Vinyl Disc Reader, by Mr. T. J. Haworth.

[8] RIAA standard quoted from page 7 of Final Year Project Report 2006, Optical Vinyl Disc Reader, by Mr. T. J. Haworth.

[9] Incredible photos of record grooves under an electron microscope - The Vinyl Factory

[10] Jim Lesurf (18 May 2000). "The 'digital' defects of the long-playing record". The University of St. Andrews. Retrieved 22 September 2017.

[11] Michael Fremer (6 January 1999). "Pass Aleph Ono phono preamplifier". Retrieved 14 May 2018.

[12] Measuring Interrupt Latency (nxp.com)

[13] Datasheet of Heason technology LA13-12-000(LTR) voice coil motor.

[14] An overview of the spec of Teensy 4.0 Teensy® 4.0 (pjrc.com)

[15] Full datasheet of NXP IMXRT106x i.MX RT1060 Processor Reference Manual (pjrc.com)

[16] Zaber T-Series Products Users Manual

[17] Jan Tommy Gravdahl, Friction Problems in Servomechanisms Modeling and Compensation Techniques, Department of Engineering Cybernetics Norwegian University of Science and Technology Trondheim

[18] Arnold Hensman, Optical Reading and Playing of Sound Signals from Vinyl Records, October 2007, The 7th IT&T Conference (ITT07), Institute of Technology Blanchardstown, Dublin, Ireland, October 2007.At: Blanchardstown, Ireland

Other readings not quoted in this report

[19] AMIR H. CHAGHAJERD, Sensing and Control in Optical Drives How to Read Data from a Clear Disc

[20] John A Cope, The physics of the compact disc

[21] Optical Vinyl Disc Player Project Notes on updated hardware Andrew Holmes, 22.02.2022

[22] Optical Vinyl Disc Player Project Suggestion on final hardware layout Andrew Holmes, 10.06.2022