ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Design Document for ECE 445

Team #34

HENGYU LIU
(hengyu2@illinois.edu)
XUAN TANG
(xuant4@illinois.edu)
HAN CHEN
(hanc7@illinois.edu)
ZHANPENG LI
(zl88@illinois.edu)

TA: Zheyi Hang

March 21, 2024

Contents

1 Introduction

1.1 Problem and Solution Overview

The advent of digital technology in music has revolutionized the way we create, perform, and interact with musical instruments. Traditional instruments, while offering rich sound and tactile feedback, often come with limitations related to size, portability, and accessibility. Instruments like the piano or double bass are not only cumbersome to transport but also require significant maintenance and careful handling due to their fragility. This presents a challenge for musicians who wish to practice or perform outside of traditional settings or for those who lack the space to house such instruments. The need for a solution that offers the tactile and auditory richness of traditional instruments, combined with the convenience and portability of digital technology, is evident.

The Virtual Band project aims to address these challenges by introducing a novel instrument interface that utilizes cutting-edge audio processing, pressure sensing, and positioning systems to convert electrical pulses into high-quality sound outputs of various instruments. Unlike previous attempts at digital musical interfaces, which often fell short in replicating the depth and nuance of real instrument sounds or the feel of playing them, our solution promises an authentic and dynamic music performance experience. By integrating motion tracking, audio processing, and sensor technology, the Virtual Band provides a portable and versatile platform for music creation through the analysis of hand movement and pressure sensitivity. This unique purpose positions the Virtual Band at the forefront of music technology innovation, showcasing a significant advancement over existing musical technologies and applications.

1.2 Visual Aid

Since we try to craft a pair of gloves with a capability to enable convenient instrument playing, which will redefine how we interact with digital environments. These gloves boast state-of-the-art 6 Degrees of Freedom (DOF) position sensors on every finger and th back of the hand, enabling unparalleled precision in capturing hand movements and poses. With a Raspberry Pi unit integrated into one glove, our design ensures stable and high bandwidth data transmission to a computer. Complementing this, an industrial-grade camera captures high-resolution videos of hand gestures, providing invaluable additional data for analysis. Our focus extends beyond mere functionality; we're committed to delivering an intuitive user experience. Through sophisticated algorithms, the computer instantly interprets and displays real-time hand poses on a user-friendly interface. But our innovation doesn't stop there. These gloves are also designed to respond dynamically, triggering tailored sounds through integrated speakers based on finger pressure sensor feedback. In essence, our project represents a leap forward in motion capture technology, promising immersive interactions and unlocking new possibilities for digital engagement.

Figure 1: Product in Context[7]

1.3 High-level requirements list

1. We can play piano music accurately, our final product can display accurate note we aim to play.
2. We can switch piano into another instrument and play music and reproduce the timber of different instrument.
3. Leveraging object detection algorithm and computation based of data transferred from sensors to accurately compute the position of fingers

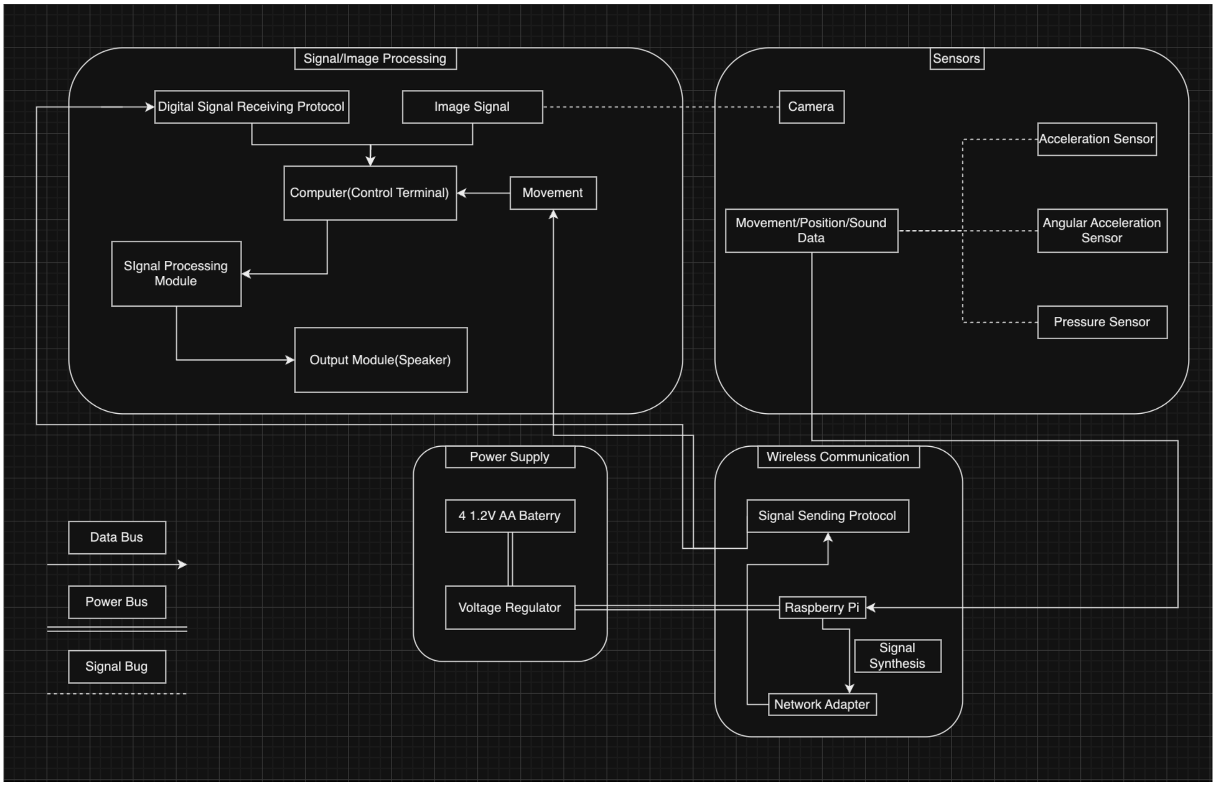


Figure 2: Block Diagram for Virtual Band[1]

2 Design

A general section looks like this. There is usually a blurb introducing the top-level section here.

2.1 Block Diagram

The Block Diagram is in Figure 2

2.2 Physical Design

This CAD file 3 illustrates the placement of key components within our Virtual Band project, providing a comprehensive overview of the installation positions for optimal functionality. The design showcases the arrangement of 6 Degrees of Freedom (6DOF) position sensors, pressure sensors, and Raspberry Pi 4B in relation to the hand, ensuring precise tracking and interaction capabilities.

Here are some figures that shows our overall design, dimensions for our micro-controller (Raspberry PI 4B) and sensors (6 DOF sensor):

Figure 3: Overall CAD Diagram for Virtual Band[4]

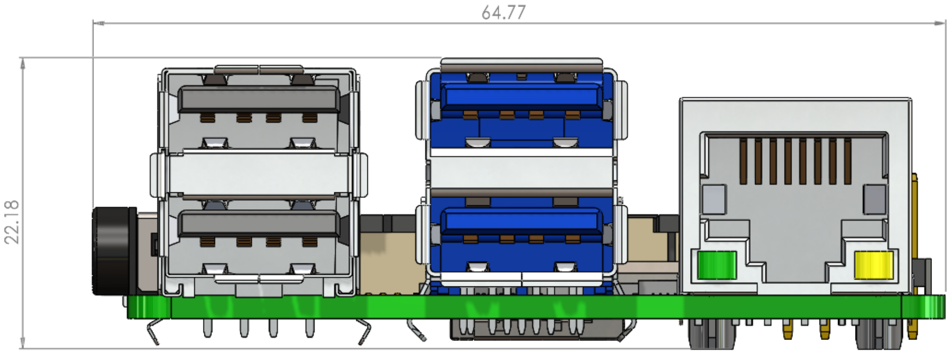


Figure 4: Raspberry Front View & Dimensions[8]

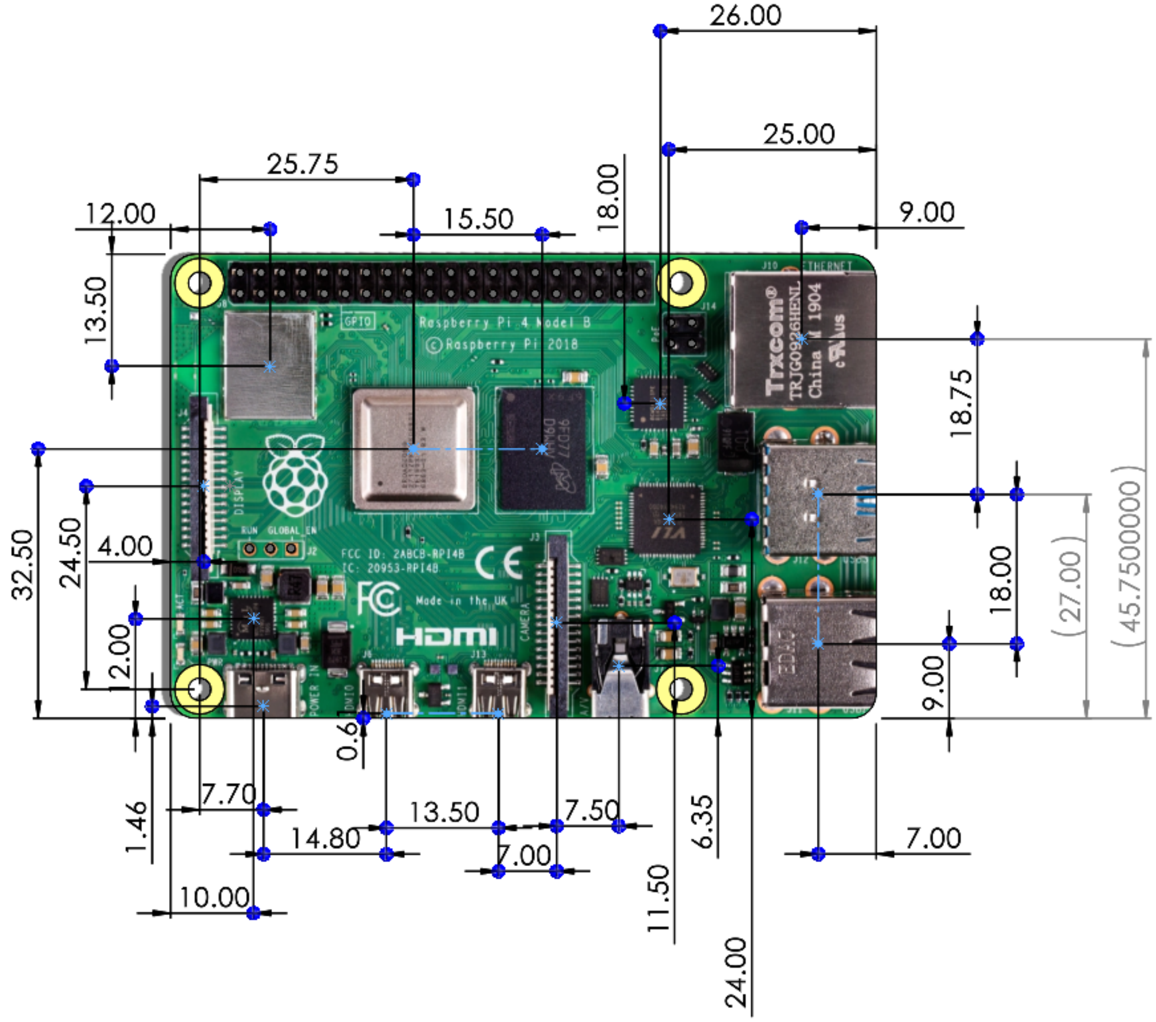


Figure 5: Raspberry Top View & Dimensions[10]

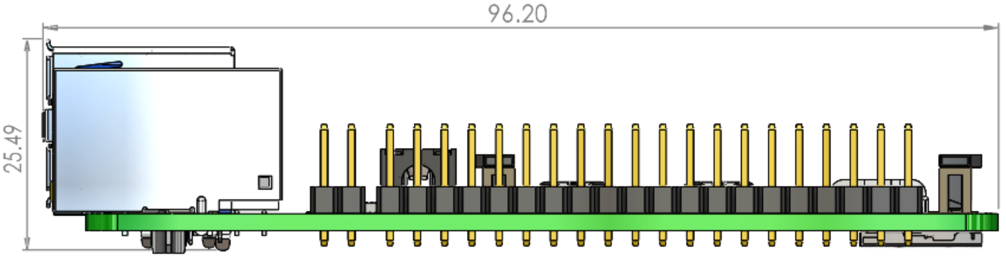


Figure 6: Raspberry Left View & Dimensions[9]

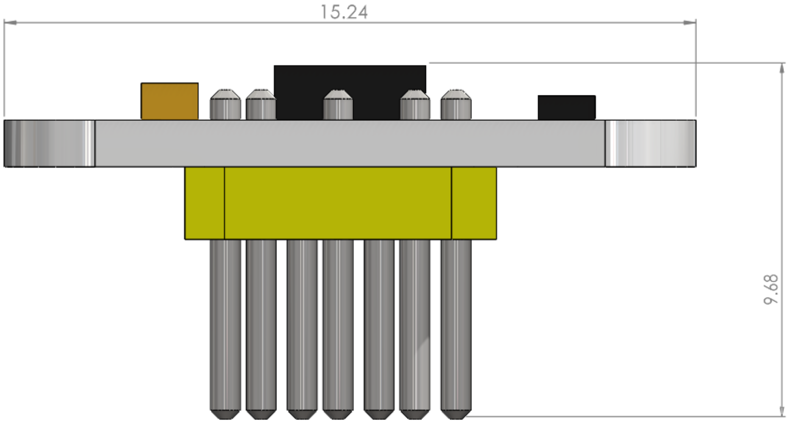


Figure 7: Sensor Side View & Dimensions[12]

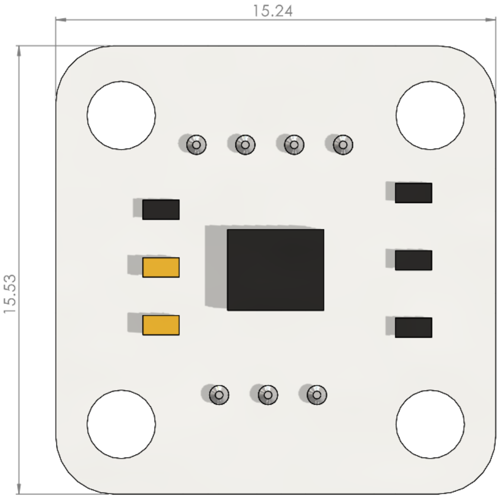


Figure 8: Sensor Top View & Dimensions[10]

2.3 Subsystems

2.3.1 Computer Vision - Finger Position Detection

The computer vision system is aiming for accurately tracking the positions of fingers wearing gloves. The system will utilize computer vision techniques and algorithms to achieve real-time tracking with high accuracy and low latency, supporting various features for virtual band performing and further modelling the keyboard keys virtually. The visual part consists of

2.3.1.1 MediaPipe Hand Landmarker

MediaPipe is an open-source framework developed by Google, designed for building multimodal (audio, video, time series, etc.) applied machine learning pipelines. It offers cross-platform, customizable ML solutions for live and streaming media. One of the notable components within MediaPipe is the Hand Tracking solution, which employs machine learning to perform real-time hand detection and tracking.

The MediaPipe Hand Landmark model detects and tracks 21 3D landmarks of a hand in real-time. This includes the position of the wrist, the knuckles, the fingertips, and the joints in between. The model utilizes a two-part process: a palm detection model that operates on the full image and proposes hand-boundaries, and a hand landmark model that operates within those boundaries to identify and track the hand landmarks.

The hand landmarker model bundle contains a palm detection model and a hand landmarks detection model. The hand landmark model bundle detects the keypoint localization of 21 hand-knuckle coordinates within the detected hand regions.

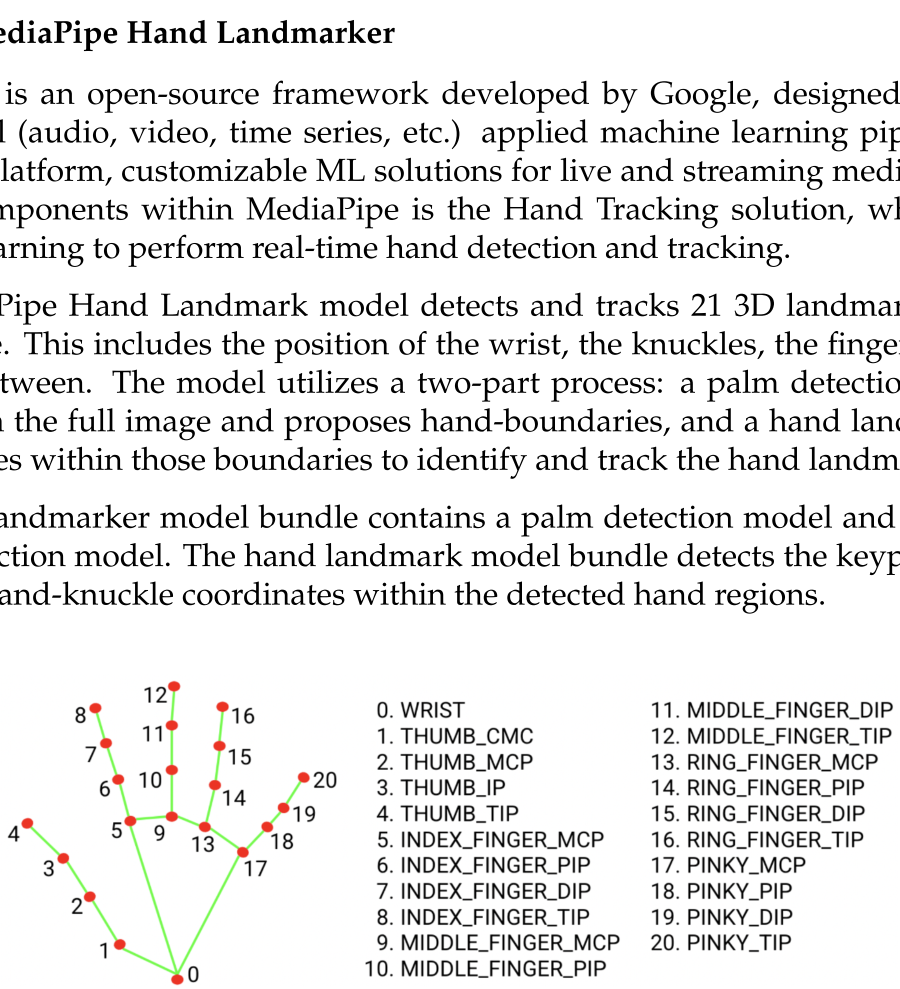


Figure 9: MediaPipe Hand Landmarks[3]

By getting the landmarks of the palm of the hand, we are able to further calculate distances between hands and the camera and model our hand fingers in specific domains. We can even detect gestures by MediaPipe model to add more features for our virtual band gloves.

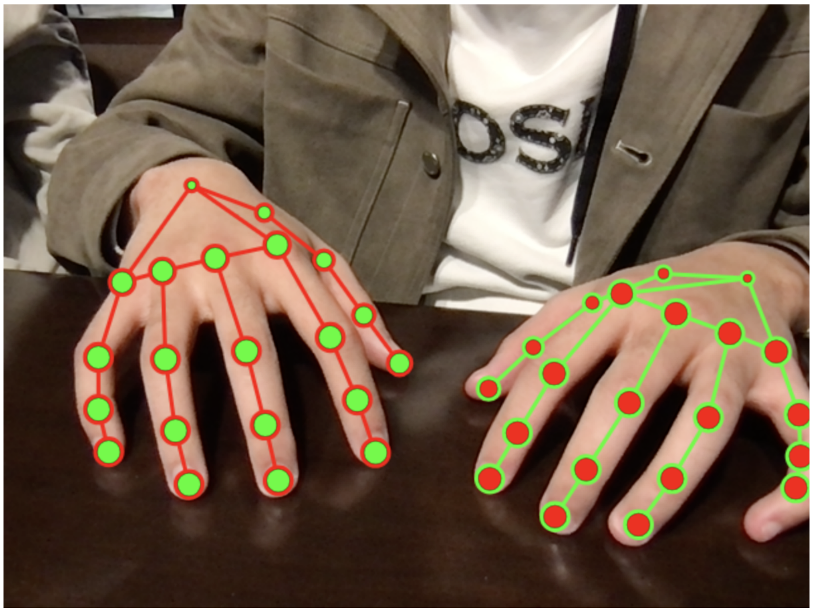


Figure 10: MediaPipe Hand Landmarker Demo[2]

2.3.1.2 Gloved Hands Detection

To bridge the gap between the appearance of bare hands and gloved hands, you can employ data augmentation techniques on your dataset of gloved hands. This involves artificially expanding your dataset with modified versions of your existing images to simulate the variety the model has been trained on.

Given that the MediaPipe hand tracking model was trained on a diverse dataset of approximately 30,000 real-world images, as well as rendered synthetic hand models superimposed on various backgrounds, adapting it to recognize gloved hands—a scenario significantly different from its training data—poses a notable challenge.

2.3.1.3 Pre-processing Gloved Hand

Since gloves can have different colors than human skin, adjusting the color space of your training images to include a broader spectrum could help. Color space adjustments involve modifying the input images in ways that alter their color properties, making the differences between various colors less pronounced or adjusting them to resemble the color distribution the model was trained on more closely. This is particularly useful when the model needs to generalize from its original training on bare hands to recognizing gloved hands, which might appear in a wide range of colors and textures.

Implementing these adjustments involves preprocessing your images before they are input into the training or inference process. By adjusting the color space of the image there's some color tunnels that would fit in the MediaPipe Model. All the landmarks is shown with the gloves on.

Figure 11: Preprocessing Color Space For Gloved Hands[6]

2.3.1.4 Retraining and Fine tuning

To bridge the gap between the appearance of bare hands and gloved hands, we employ data augmentation techniques on our dataset of gloved hands. This involves artificially expanding our dataset with modified versions of our existing images to fine tune the model of the mediapipe hand landmarker.

Retraining a model using Model Maker generally makes the model smaller, particularly if we are trying to train for gloved hands with our PCB and equipment sensors on which is a specific task. Retraining a model with Model Maker cannot change what the original ML model was built to solve, even if those jobs are similar.

2.3.1.5 Datasets

Palm and gloves dataset for Object Detection or AR systems

The dataset consists of images of Human palm captured using mobile phone. The images have been taken in real-world scenario like holding objects or performing simple gestures. The dataset has wide variety of variations like illumination, distances etc. It consits of images of 3 main gestures: Frontal-open palm, Back open palm and fist with wrist. It also have a lot of images with people wearing gloves.

Requirement	Verification
Ensure accurate hand landmark positioning	Check whether the landmarker are shown correctly
Since our hands would be gloved and features with multiple equipments, normal mediapipe model could not handle landmarker tasks	make sure gloved hands can be fit in our retrained model
Annotate the palm gloved hand datasets and preprocessing based on color space adjustment methods	Retraining a model with Model Maker using self-annotated datasets.

Table 2: Requirements and Verification (R&V) Table for this submodule

2.3.2 Virtual modeling - Building a model of piano key positions

In this part of the computer vision modeling, we plan to use the positioning and modeling of the camera to assist the positioning of the sensor to the key, so as to obtain more accurate positioning and key differentiation. In addition, we also need to complete the audio processing, including the different audio feedback of the tap and the heavy press, the audio length of the long press and the short press, the pitch of the different keys. This module contributes to the overall design by facilitating accurate key positioning and differentiation. It enhances user experience by providing visual aids for locating keys and ensures precise fingertip positioning. The handshake debugging session improves system reliability by verifying the consistency between pixel movement and actual distances.

2.3.2.1 Interface

Input: Images captured by the camera, Pressure levels from the pressure sensor.
Medium Output: Pixel coordinates of fingertips and key positions, Key positions determined by the Camera
Output: Audio feedback corresponding to key presses.

2.3.2.2 Camera and Laser Positioning Module

To address the immediate requirement of accurately locating the standard tone, our primary focus will be integrating an auxiliary sensor alongside the existing camera system. By incorporating a small laser module adjacent to the camera, users will be provided with a precise reference point for identifying the key corresponding to the standard tone. This addition enhances user experience by offering a visual cue alongside the auditory feedback.

Simultaneously, leveraging the capabilities of the hand recognition module will facilitate fingertip positioning, further enhancing the accuracy and ease of use of the system. By integrating this functionality into the existing framework, users will have the ability to pinpoint specific notes with precision, thus streamlining the calibration process.

Upon successful implementation of the aforementioned features, our next step will involve conducting a handshake debugging session. This session aims to validate the accuracy of finger movement tracking by correlating pixel movement within the camera's field of view with the actual physical distance measured by the sensor. By establishing this correspondence, we ensure the reliability and consistency of the system across different hand movements and positions.

In summary, the proposed design enhancements prioritize the integration of an auxiliary sensor and laser module for precise tone localization, complemented by hand recognition technology for enhanced fingertip positioning. Subsequent validation through handshake debugging will affirm the accuracy and robustness of the system, ensuring optimal performance for users.

2.3.2.3 Audio Processing Module

In the realm of audio processing, our primary objective revolves around correlating the determined position, derived from sensor data, with the corresponding key audio output. A pivotal challenge lies in effectively integrating pressure levels from the sensor with audio volume levels. Our proposed solution entails a systematic approach:

Pressure-Resistance Curve Calibration:
Initially, we undertake the task of measuring the sensor responses across varying pressure levels, subsequently constructing a pressure-resistance curve. This empirical data serves as the foundation for our calibration process.

Curve Fitting and Inverse Function Utilization:
Employing curve-fitting techniques, we derive a mathematical model that accurately represents the relationship between pressure and resistance. By applying the inverse function to this model, we can efficiently convert voltage readings from the sensor into corresponding pressure values.

Volume Adjustment Based on Pressure:
Leveraging the pressure data obtained, we dynamically adjust the volume of the audio output. This adjustment ensures that the audio volume aligns appropriately with the pressure exerted on the sensor, providing users with an intuitive and responsive auditory experience.

Handling Concurrent Key Presses:
In scenarios where multiple keys are pressed simultaneously, we adopt a provisional approach wherein audio playback for each pressed key occurs concurrently. This interim solution aims to fulfill project requirements efficiently. However, we remain attentive to potential limitations in user experience and explore avenues for improvement.

Potential Digital Signal Processing (DSP) Enhancement:

Should the concurrent playback approach yield unsatisfactory results, we remain open to implementing digital signal processing techniques to refine sound wave interactions. This step would involve sophisticated algorithms aimed at mitigating interference and optimizing audio output quality in multi-key press scenarios.

In summary, our approach emphasizes the calibration of pressure-resistance curves, leveraging mathematical models to facilitate pressure-to-volume conversion, and accommodating concurrent key presses through simultaneous audio playback. Furthermore, we remain receptive to the possibility of integrating advanced DSP techniques to enhance audio fidelity and user satisfaction.

Requirement	**Verification**
Ensure accurate positioning and modeling of the camera for precise sensor alignment	Check that the tracking points are at the right places.
Provide audio feedback corresponding to different key presses	Verify audio feedback correctness against key presses.
Integrate an auxiliary laser module for precise key localization & Implement hand recognition for enhanced fingertip positioning.	Be able to locate the center point and receive the feedback of calibration.
Conduct handshake debugging session to ensure accurate finger movement tracking.	The program is able to record and compare the distance with the number of pixels on screen.
Utilize mathematical models for pressure-to-volume conversion & Implement concurrent audio playback for multiple key presses.	Validate pressure-to-volume conversion accuracy through controlled pressure exertion and audio volume comparison.

Table 4: Requirements and Verification (R&V) Table for this submodule

2.3.3 Hardware Integration - Raspberry Pi 4B & Sensors

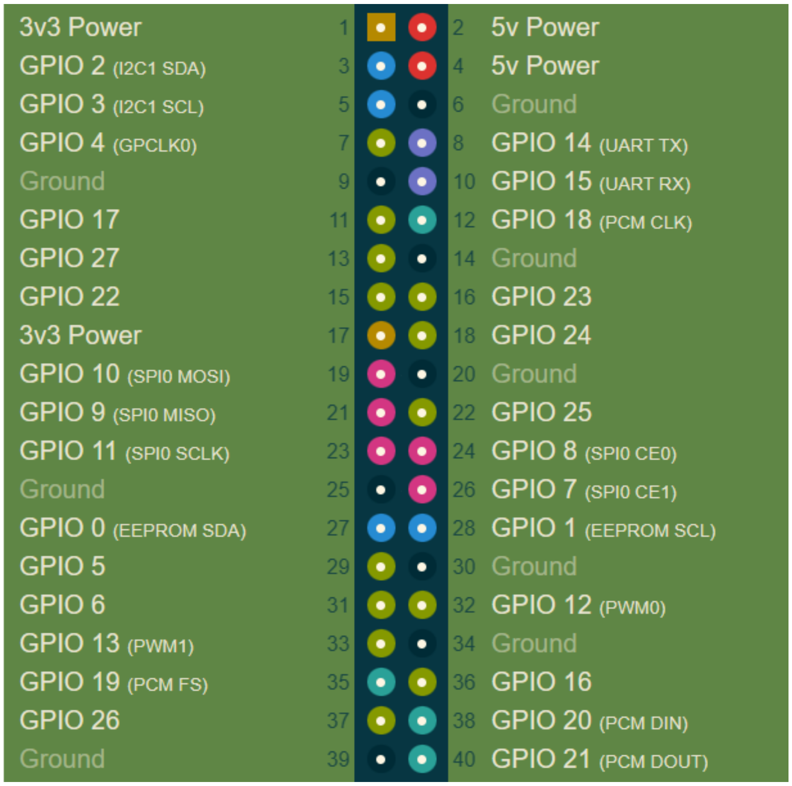
2.3.3.1 Microcontroller (Raspberry PI 4B)

Input: 5V power supply, 3-axis translational acceleration and 3-axis angular acceleration data, voltage measurement for pressure, video stream from the camera
Output: Pressure sensor and acceleration sensor data

This unit is mainly used to pull the 3-axis translational acceleration and 3-axis angular acceleration data out of GY 521 - MPU 6050 V2 or JY931 6 degree of freedom sensors, and combine it with the pressure data collected by RP-C7.6-LT-LF2. These data will be processed within Raspberry PI and sent to the computer through signal synthesis and network. Data transfer and communication will be further discussed within "Module Communication and Data Transfer" section.

For our project, the Raspberry PI 4B will be powered by a +5V voltage source regulated from two +3.7V 18650 batteries connected in parallel.

The pin layout of the Raspberry PI 4B is displayed in Figure 12



3v3 Power	1	2	5v Power
GPIO 2 (I2C1 SDA)	3	4	5v Power
GPIO 3 (I2C1 SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART TX)
Ground	9	10	GPIO 15 (UART RX)
GPIO 17	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	20	Ground
GPIO 9 (SPI0 MISO)	21	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	24	GPIO 8 (SPI0 CE0)
Ground	25	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)

Figure 12: Pin Layout for Raspberry PI 4B[5]

Requirement	Verification
The module should accurately capture and process 3-axis translational acceleration and 3-axis angular acceleration data from sensors such as GY 521 - MPU 6050 V2 or JY931 6-degree of freedom sensors.	Validate the accuracy and reliability of 3-axis translational acceleration and 3-axis angular acceleration data acquisition from the connected sensors.
It must include functionality to measure voltage for pressure data collected by the RP-C7.6-LT-LF2 pressure sensor.	Verify the module's ability to measure voltage accurately for pressure data collected by the RP-C18.3-LT pressure sensor.
The module should be capable of receiving and processing a video stream from the camera connected to the Raspberry Pi 4B.	Confirm that the module can successfully receive and process a video stream from the connected camera, ensuring smooth operation without lag or errors.
It should synthesize and combine the collected sensor data and video stream data within the Raspberry Pi.	Test the module's functionality in synthesizing and combining sensor data and video stream data within the Raspberry Pi.

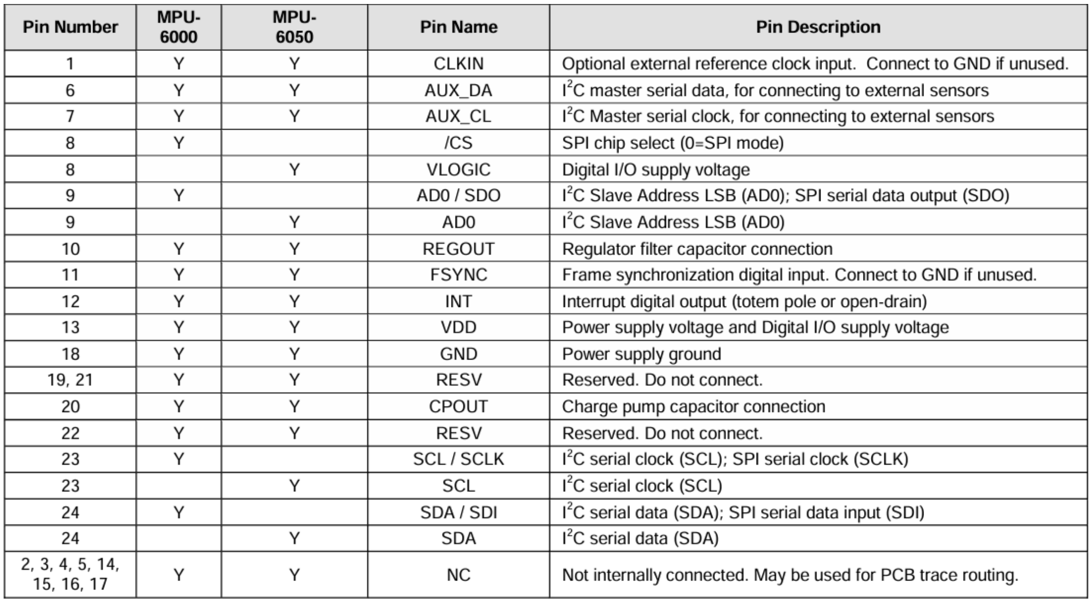
Table 6: Requirements and Verification Table for Power Supply

2.3.3.2 6DOF Position and Pose Sensor

Input: Physical translation and orientation of the sensor, 3.3/5V voltage source
Output: Digitalized translational and angular acceleration data

For accurate position and pose sensing, we decide to use either MPU6050 or JY931, while they share similar precision and accuracy. The desirable range scale for translational acceleration would be ±8wg, and ±10 rad/s^2, with precision of 0.1g and 0.1 rad/s^2.

The MPU6050 (LQ6050MV2) module features the original MPU-6050 sensor chip and offers versatile installation options with two types of pin headers: straight (7P) and bent (7P). It operates with either a 3.3V or 5V power supply, facilitated by internal low dropout voltage regulators. Utilizing standard IIC communication protocols, it incorporates a 16-bit converter for high-resolution data output. The gyroscope section provides 131, 65.5, 32.8, and 16.4 LSB/(°/s) sensitivity options corresponding to ±250, ±500, ±1000, and ±2000°/sec, enabling precise angular velocity sensing across three axes. Programmable control extends to the accelerometer section, offering selectable ranges of ±2g, ±4g, ±8g, and ±16g, with corresponding 16384, 8192, 4096, and 2048 LSB/g. Additionally, the module supports programmable interrupts for various functions, including posture recognition, shake detection, image scaling, scrolling, high-G sensing, motion detection, tap sensing, and shake sensing. With a pin spacing of 2.54mm (100mil) and constructed using immersion gold FR4 PCB technology with machine soldering processes, the MPU6050



Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I^{2}C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I^{2}C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip select (0=SPI mode)
8		Y	VLOGIC	Digital I/O supply voltage
9	Y		AD0 / SDO	I^{2}C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I^{2}C Slave Address LSB (AD0)
10	Y	Y	REGOUT	Regulator filter capacitor connection
11	Y	Y	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	Y	Y	INT	Interrupt digital output (totem pole or open-drain)
13	Y	Y	VDD	Power supply voltage and Digital I/O supply voltage
18	Y	Y	GND	Power supply ground
19, 21	Y	Y	RESV	Reserved. Do not connect.
20	Y	Y	CPOUT	Charge pump capacitor connection
22	Y	Y	RESV	Reserved. Do not connect.
23	Y		SCL / SCLK	I^{2}C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I^{2}C serial clock (SCL)
24	Y		SDA / SDI	I^{2}C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I^{2}C serial data (SDA)
2, 3, 4, 5, 14, 15, 16, 17	Y	Y	NC	Not internally connected. May be used for PCB trace routing.

Figure 13: Pin Layout for MPU6050

ensures reliable performance and quality assurance.

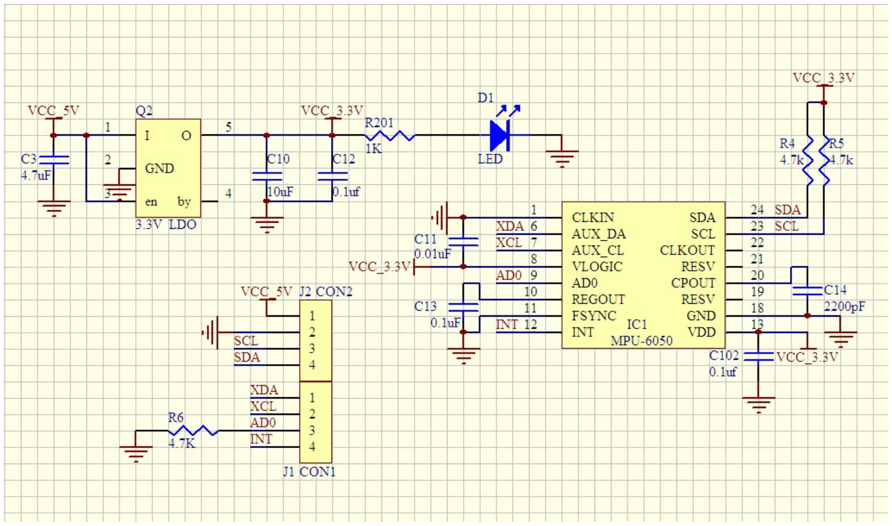


Figure 14: Schematic of MPU6050**[fig˙pin˙layout˙mpu6050]**

The pinout for MPU6050 is displayed in Figure 13 and schematic of MPU6050 is displayed in Figure 14.

Requirement	Verification
The sensor must accurately measure translational acceleration within the range of ±8g with a precision of 0.1g.	Apply known translational acceleration values within the specified ranges to the sensor and verify the accuracy of the measured values against the expected values.
It should accurately measure angular acceleration within the range of ±10 rad/s^2 with a precision of 0.1 rad/s^2.	Apply known angular acceleration values within the specified ranges to the sensor and verify the accuracy of the measured values against the expected values.
The sensor should support programmable control for adjusting sensitivity and range settings according to application requirements.	Test the sensor's programmable control features by adjusting sensitivity and range settings programmatically and verifying the corresponding changes in output values.
It must offer programmable interrupts for various functions such as posture recognition, shake detection, image scaling, scrolling, high-G sensing, motion detection, tap sensing, and shake sensing.	Confirm the functionality of programmable interrupts by triggering various interrupt conditions and observing the sensor's response.

Table 8: Requirements and Verification Table for Power Supply

2.3.3.3 Pressure Sensor

Input: Physical pressure applied on the finger tip
Output: Pressure sensor and acceleration sensor data

As we have mentioned before, the sound level of the instrument is determined by the data collected by the pressure sensor. The measurable range should be 0g - 1kg with precision of 1g.

This pressure sensor features a slim design with a thickness of 0.3mm, making it ideal for applications requiring space efficiency. It activates with a trigger force of 30g, signified by a default resistance of less than 200kΩ. Capable of measuring pressures within the range of 2g to 1.5kg, it accommodates both static and dynamic pressure measurements at frequencies up to 10Hz. With an initial resistance exceeding 10MΩ, it swiftly responds to pressure changes in less than 0.01s. Operating seamlessly across temperatures ranging from -40°C to +85°C, it offers long-lasting performance with a lifespan exceeding one million cycles. This sensor maintains excellent consistency, with resistance variations of +/-3% within individual units and +/-10% across batches under equivalent testing conditions. Additionally, it exhibits minimal hysteresis, with a drift of less than 5% after 24 hours under a static load of 1kg. Furthermore, it demonstrates immunity to electromagnetic interference (EMI) and electrostatic discharge (EDS), ensuring reliable operation in various environmental conditions.

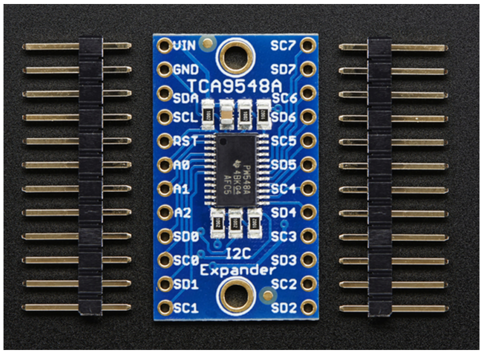


Figure 15: TCA9548A Pressure Sensor[**fig˙tca9548a˙pressure˙sensor**]

The pressure sensor will be integrated into the circuitry with a resistor connected in series. The voltage output from the pressure sensor, reflecting the detected pressure, will be directed to the TCA9548A multiplex ADC. This ADC is equipped with eight channels for digitalization, enabling simultaneous conversion of analog signals from multiple sensors. The TCA9548A will facilitate communication between the pressure sensor and the Raspberry Pi microcontroller using the I2C (Inter-Integrated Circuit) protocol. Through this communication protocol, the Raspberry Pi will receive digitalized pressure data from the ADC, allowing for further processing and analysis within the system. This setup ensures efficient and accurate transmission of pressure information from the sensor to the Raspberry Pi for real-time monitoring and control applications.

Requirement	**Verification**
The pressure sensor must accurately measure pressures within the range of 0g to 1kg. The precision of pressure measurement should be 1g.	Apply known pressure values within the range of 0g to 1kg to the sensor and verify the accuracy of the measured pressure values against the expected values.
The sensor should activate when a trigger force of 2g is applied, indicated by a default resistance of less than 200kΩ.	Verify the sensor's sensitivity and threshold by applying incremental pressure changes within the measurable range and observing the corresponding output changes.

Table 10: Requirements and Verification Table for Power Supply

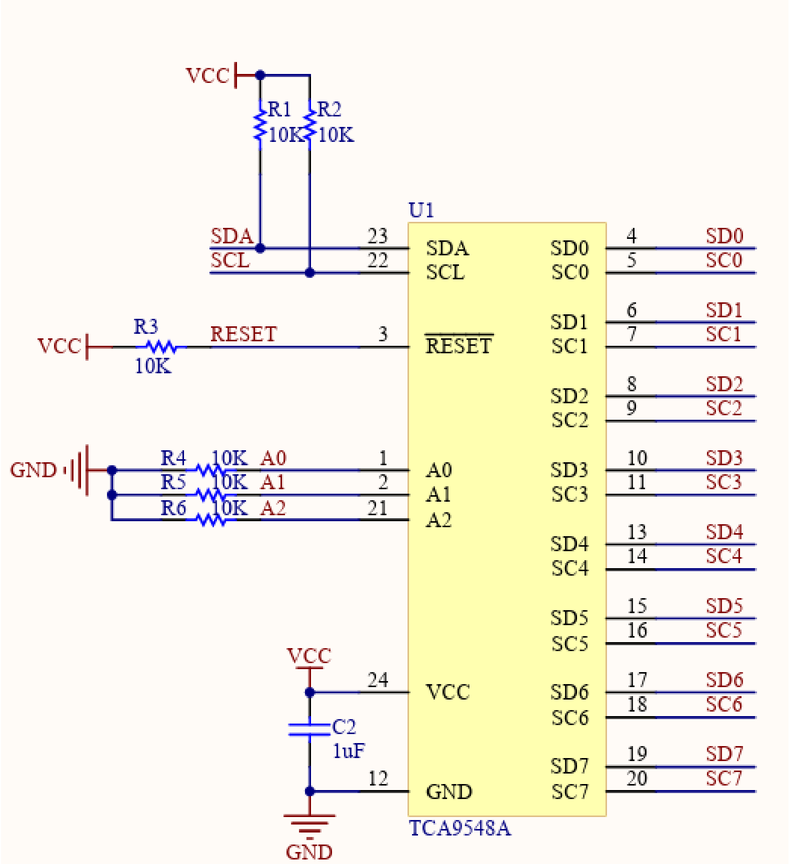


Figure 16: Schematic of TCA9548A[11]

2.3.4 Power Supply - Battery & Regulator

Input: 2 18650 Battery
Output: Pressure sensor and acceleration sensor data

We choose to use 18650 Raspberry Pi dedicated power module as battery source for our project. It is a versatile solution designed to provide stable and reliable power to your Raspberry Pi projects. With its robust features and flexible design, this power module offers an efficient and convenient way to power your Raspberry Pi devices while ensuring optimal performance and extended runtime.

At the heart of this power module are two high-capacity 18650 lithium-ion batteries, each boasting a capacity of 6800mAh. These batteries are configured in parallel, providing a total capacity of 6800mAh for prolonged usage without the need for frequent recharging. This power module is able to deliver a consistent 5V-5.1V/3A output voltage to our Raspberry Pi. This ensures reliable and uninterrupted power supply, essential for maintaining the stability and performance of our projects. This includes the power supply to the Raspberry PI, position and pose sensors, pressure sensors, and a analog-to-digital converter.

Furthermore, the power module boasts three USB-A interfaces capable of delivering a 5V 3A power output each, providing ample power for various peripherals and devices. Additionally, it features a micro-B and a type-C interface, both capable of delivering a 5V 2A power output. This allows us to power additional we have with ease.

Requirement	**Verification**
The power module must provide a stable output voltage within the range of 5V-5.1V.	Use a digital multimeter (DMM) to measure the output voltage of the power module.
The voltage output should be capable of delivering a current of up to 3A to the connected devices.	Apply a load equivalent to 3A to verify the voltage stability under maximum load conditions.

Table 12: Requirements and Verification Table for Power Supply

2.3.5 Module Communication and Data Processing

This part contains several key components: Respberry PI 4b, Sensors, camera and ONE computer.

2.3.5.1 Sensor Integration

The subsystem is tasked with the integration of 22 sensors distributed across both hands of the musician, comprising 5 pressure sensors and 6 acceleration sensors (JY931) per hand. Each acceleration sensor outputs data in 6 Degrees of Freedom (DOF), with each DOF data being 4 bytes, and operates at a frequency of 1000 Hz. The high-frequency operation is pivotal for capturing the nuanced movements of a musician's hands, translating these into precise musical expressions.

2.3.5.2 Noise Filtering and Data Preprocessing

Given the sensitivity of the sensors and their susceptibility to noise, the subsystem incorporates advanced filtering algorithms executed within the Raspberry Pi 4B modules. These algorithms are designed to eliminate irrelevant or noisy signals, ensuring that only meaningful data is transmitted for further processing. This filtering is particularly crucial for pressure sensors, where even slight pressure variances induced by the glove's material must be accurately accounted for to avoid distortion of the intended musical input. To mitigate the effects of noise and enhance signal clarity, the subsystem employs sophisticated filtering algorithms within the Raspberry Pi 4B modules. These algorithms are tailored to the specific noise profiles and operational frequencies of the pressure and acceleration sensors, leveraging technique. Instead of transmitting continuous sensor data, which includes a significant amount of redundancy and irrelevant information, the system focuses on sending peak data. This approach significantly reduces the amount of data that needs to be transmitted, focusing on those data points that accurately represent the musician's intent.

2.3.5.3 Communication Protocol Design

This subsystem leverages a dual-protocol strategy to optimize data transfer:

HTTP Protocol: Ensures data integrity and is used for intra-network communication. This protocol is particularly suited for environments where network consistency can be maintained, providing a reliable channel for data transfer. gRPC: Facilitates faster data transfer rates, essential for reducing latency in musical performance. By employing gRPC, the subsystem ensures that data packets are efficiently encoded and transmitted across network boundaries, enabling real-time interaction between the sensors, Raspberry Pi modules, and the central laptop. Both protocols operate through four dedicated ports on each device, enhancing data security and ensuring that the data streams remain segregated and protected.

2.3.5.4 Synchronization and Consistency

Addressing the Byzantine Fault Tolerance (BFT) problem, the subsystem integrates a set_time function across all components to mitigate time drift and ensure synchronized operations. This function is invoked every minute, aligning the internal clocks of the sensors, Raspberry Pis, and the central processing unit. By embedding timestamps within the transmitted data packets, the system can correlate data streams with precision, ensuring that simultaneous events across the system are processed coherently.

2.3.5.5 Sensor and Camera Data Processing

Processing data from multiple sensors and a camera to accurately capture and interpret a musician's movements and inputs involves a sophisticated integration of sensor data, algorithmic decision-making, and error correction methodologies. At the core of this process is the objective to meticulously translate physical actions into precise digital outputs that reflect the intended musical expressions on a virtual piano interface.

The operational logic begins with data acquisition from acceleration sensors attached to each finger and hand of the player. These sensors are critical for determining the relative movement of the fingers from a predefined central zero point. By analyzing this movement data, the system can infer which keys on the virtual piano are being pressed or intended to be pressed by the musician. This inference is based on a pre-established piano model that maps specific spatial coordinates to corresponding piano keys, thus allowing for the accurate translation of physical movements into specific musical notes.

Concurrently, visual data captured by the camera serve a dual purpose: enhancing the accuracy of movement detection and providing a secondary verification mechanism to ensure the reliability of the sensor data. The camera monitors the positioning of the fingers in real-time, offering an external perspective that is particularly valuable in cases where the acceleration sensors might encounter errors or inaccuracies—especially when the musician's hands move rapidly or far from the central zero point.

To reconcile any discrepancies between the data derived from the acceleration sensors and the visual confirmation from the camera, the system employs a weighted harmonic average algorithm. This sophisticated algorithm takes into account both sets of data, prioritizing accuracy by adjusting for potential sensor errors or visual misalignments. By calculating the weighted harmonic average of the two positions, the system effectively harmonizes the sensor and camera data, ensuring that any deviation or error is corrected and that the determined finger position is as accurate as possible.

Once the system has confidently identified the finger positions and detected a corresponding signal from the pressure sensors—indicating not just contact with a virtual key but also the intensity of the press—it proceeds to generate the appropriate musical note. This note is then sustained for the duration of the signal received from the pressure sensors, mirroring the natural behavior of a piano where the sound persists for as long as the key is pressed.

Requirement	**Verification**
Latency must be less than 70ms for real-time performance	A. Perform end-to-end system latency tests using timestamped sensor data. B. Verify latency under maximum load using a network analyzer to eliminate latency by re-write submodules that may cost large latency.
Data transfer bandwidth must accommodate the sensor data rate	A. Calculate minimum required bandwidth: $11 \times 2 \times 6 \times 4 \times 1000 = 528000$ bytes/sec. B. Test with network monitoring tools to ensure bandwidth exceeds this rate. Also make sure PI can afford this large data load in each pin.
Sensor data must be accurate to within a specified threshold, for pressure sensor, can detect 0.02N to 15N, and for acceleration sensors, accuracy must be in 6 digits	A. Calibrate sensors against a known standard. B. Test sensor output by simulating controlled movements and compare against expected data.
Noise filtering must effectively isolate and remove unwanted signals	A. Introduce known noise patterns into sensor data, such as steady noisy caused from gloves. B. Verify filtering algorithms remove them and analyze S2N ratio.
Synchronization across all system components must be maintained within a small margin of error	A. Implement $'set\_time'$ function. B. Verify synchronization with long-duration testing to ensure minimal time drift.

Table 14: Requirements and Verification (R&V) Table for this submodule

2.4 Tolerance Analysis

2.4.1 Tolerance Analysis for Computer Vision Task

The tolerance analysis reveals significant insights into the performance and fairness of the MediaPipe hand tracking model, highlighting areas for improvement and further research to ensure equitable and accurate landmark detection across all user groups.

700 images, 50 images from each of the 14 geographical subregions, 420 images, 35 images from each unique combination of the perceived gender and the skin tone (from 1 to 6) based on the Fitzpatrick scale. The evaluation results across six skin tone types, based on the Fitzpatrick scale, show that both models tend to perform better on certain skin tones than others. The range of MNAE across skin tones for both lite and full models suggests potential bias in landmark detection accuracy related to skin tone. The models' performance also varies between genders, albeit with a smaller range of MNAE.

Analysis by Skin Tone

- For the **lite model**, the average MNAE across six skin tone types is $5.67\% \pm 0.94\%$ with a range of $[4.88\%, 7.25\%]$, indicating variability in the model's performance across different skin tones.
- The **full model** shows a slightly better average performance of $5.08\% \pm 0.72\%$ with a range of $[4.53\%, 6.21\%]$, suggesting more consistency across different skin tones, likely due to a more complex architecture or additional training data.

Analysis by Gender

- The **lite model's** average MNAE for gender evaluation is 5.67%, with a performance range between 5.29% and 6.05%. This range suggests fairly consistent performance across genders, though the higher end indicates room for improvement.
- The **full model** improves average performance to 5.09% MNAE, with a range of 4.80% to 5.38%, demonstrating more consistent performance across genders.

Key Takeaways:

Both models show reasonable consistency across skin tones and genders, with opportunities for further refinement highlighted by variations in MNAE. Enhancements in the training dataset's diversity and specific model tuning for underperforming groups may improve overall fairness and effectiveness.

In addition, in the positioning task, we plan to set the width of the white key to about 2.3 cm to 2.8 cm, and the black key to about 9.5 cm to 1.5 cm. Due to the error caused by the sensor accuracy and camera distortion, we increase the error range of 2 to 3 mm. If a position is in a blur between two keys, we choose the note near the center of the piano (reference note), because the farther away from the center of the hand distance error is greater.

2.4.2 Tolerance Analysis for submodule Module Communication and Data Processing

The system's accuracy and real-time responsiveness are dependent on the precision and reliability of multiple integrated sensors and communication protocols. Tolerance analysis becomes a crucial aspect of the system design to ensure seamless functionality.

Pressure sensors in the system are capable of detecting forces ranging from 1N to 50N. We assume a nominal tolerance of $\pm 0.5N$ for these sensors, which is reasonable considering the dynamic range and application. Acceleration sensors require a high degree of precision, and for the purposes of this analysis, a tolerance of $\pm 0.0001g$ is deemed acceptable, with g representing standard gravitational acceleration.

Data communication is facilitated through Raspberry Pi 4B modules, which handle sensor data at a frequency of 1000 Hz, with each degree of freedom (DOF) producing 4 bytes of data. To sustain real-time data transfer without perceivable latency for musical performance, we provision a bandwidth tolerance of 1Mbps above the computed requirement of 528000 bytes/sec, factoring in a 95% efficiency rate to accommodate protocol overhead.

Noise filtering is imperative for the system, with algorithms in place to maintain a signal-to-noise ratio (SNR) above 20dB. This ensures that the nuances of the musician's movements are captured accurately. The system's noise tolerance must not exceed levels that would compromise the SNR beyond this benchmark.

Synchronization tolerances across the system's components are set to a maximum permissible time drift of $\pm 5ms$ over a 1-minute interval. This specification is critical to maintain the timing and coherence of the musical output, ensuring a synchronous and harmonious performance.

3 Cost and Schedule

3.1 Cost Analysis

- Labor:
  - For each partner in the project, \$20/hour as the standard salary, everyone works at least 6 hours/week
  - Use the formula $(\$20/\text{hour}) \times 2.5 \times (9 \text{ weeks} \times 6 \text{ hours/week}) = \text{TOTAL}$
  - Add details about the total labor for all partners here.
- Parts:
  - table listing all parts (description, manufacturer, part number, quantity and cost)
    Every component below is for one hand. If we have extra time after succeeding in finishing the right hand, we will buy another set for the left hand. Every component will double.

Num	Part (for one hand)	Item	Cost
1	GY-521 MPU6050 module 3D Angle sensor 6DOF three-axis accelerometer six-axis gyroscope	1	¥161.24
2	Weite intelligent serial port nine-axis high-precision accelerometer gyroscope module attitude Angle sensor JY931	5	¥142
3	Raspberry PI 4B 8g	1	¥549
4	Shida anti-cut gloves	1	¥35.73
5	PCA9548 PCA9548A 1-to-8 I2C 8-way IIC multiway expansion board Module development board	1	¥5.28
6	Raspberry PI 4B/3B+ 18650 Battery expansion board UPS uninterruptible power supply 5V3A power supply lithium battery module	1	¥107
7	Thin film pressure sensor RP-C FSR402 406 Flexible resistive force and pressure sensitive plantar-robot	5	¥87.5
8	Dupont thread	3	¥29.94
9	Breadboard 400 Tie-Points	3	¥9.72

Table 16: Cost Analysis

- Sum of costs into a grand total:
  - total grand is $1000

3.2 Schedule

Date	**Xuan Tang**	**Hengyu Liu**	**Han Chen**	**Zhanpeng Li**
3/11	Set up environment for yolo v9 recognition Implemented yolo v9 for capturing hand position and motion information	Started initialization of the raspberry PI 4B and try to set up the development environment.	For capturing the hand position, assuming we have identified the finger position, how to calculate the distance is the problem I try to solve.	Completed purchase of the sensors including 6DOF sensors with different precision (GY-521 and JY931) and microcontrollers (Raspberry PI 4B).
3/18	Complete code analysis and schedule, discuss detailed implementation for finger position detection.	Complete high-level requirement in design document, set up VNC viewer for developing purpose on raspberry pi	Try to get formula to calculate the distance and determine piano key. Finish tolerance analysis and formula calculation in design document.	Complete Visual Aid and Physical Design for the product.
3/25	Collect and annotate datasets for palm and gloved hands detection, evaluate performance of MediaPipe model on gloved hands	Complete data transfer protocol implementation bewteen sensors and rasberry pi	Test code that displays and realizes the function of piano key. Write the protocol from existing code to for later signal processing.	Test and validate force vs pressure sensor curve.

4/1	Testing the performance of color space adjustment on gloved hands	Try to connect the position and pose sensor with the raspberry pi and be able to read the data, try to connect the pressure sensor with the raspberry pi and be able to read the data	Find some other instruments database. Help the team connect all sensor and get the output signals. Discuss the method of treating signals.	Connect pressure sensors to 8-channel I2C multiplexer, Design a connection part for I2C multiplexer if necessary. Assess the need for a connection part based on the setup and requirements
4/8	Evaluate the color space adjustment method performs bad in gloved hands detection, retrain the model using model maker	Complete latency analysis,if the latency go over 70ms, adjust the protocol and data processing process, aiming to decrease latency of concurrency actions.	Connect signal processing parts with notes playing. (being able to play notes when pressing)	Design a connection part compatible with the multiplexer and other components. Prototype the design and test its functionality. .
4/15	Evaluate the retrained model for the gloved hands detection, test on our virtual band gloves	Integrate data transfered from all fingers, and adjust the transfer bandwidth and protocol	Finish locating module. Finish distance calculation and key determination.	Prepare the position and pose sensors for testing. Develop a testing procedure to assess the functionality and accuracy of the sensors.

4/22	Debug and optimize the landmark detecting model for real-time detection	Integrate camera and sensors for data processing, test audio protocol and nois cancelation method. Address the set_time function to solve the consensus problem	Finish location adjustment and calibration parts.	Conduct tests to evaluate the performance of the sensors under different conditions. Iterate on the design if necessary to ensure compatibility and functionality.
4/29	Complete Vision task for our specific gloved-hand landmark detection	Debug and Analyze test results and identify issues or areas for improvement	Test play accurate keynotes.	Debug and Analyze test results and identify any issues or areas for improvement
5/6	Test and demo, self evaluation and beautify design	Test and demo, self evaluation and beautify design	Test and demo, self evaluation and beautify design	Test and demo, self evaluation and beautify design

Table 19: Schedule of Tasks

4 Discussion of Ethics and Safety

4.1 Ethics

In developing the Virtual Band Model Project, our team is committed to adhering to the highest ethical standards as outlined by the IEEE/ACM Code of Ethics. We recognize the importance of ethical responsibility in the development and implementation of our technology, particularly given its innovative integration of hardware and software components to create a virtual musical experience.

4.1.1 Professional Responsibility

We commit to making decisions and taking actions that are in the best interests of society, public safety, and the environment. This involves ensuring that our power supply module, signal processing module, sensor module, and wireless communication module are designed and tested to prevent harm. For instance, our voltage regulator and battery design considerations prioritize user safety and device reliability, avoiding potential overvoltage or undervoltage situations that could lead to damage or injury.

4.1.2 Quality and Reliability

We pledge to uphold the quality and reliability of our system by rigorously testing our components, such as the GY-521 PU6050 sensors and FSR 402 pressure sensors, to ensure they meet specified requirements for precision and force detection. By doing so, we safeguard against inaccuracies that could affect the user experience or cause unintended harm.

4.1.3 Privacy and Data Protection

Our project incorporates the use of cameras and sensors to detect user movements and interactions. We are committed to protecting the privacy of users by implementing robust data handling and storage protocols. Personal data collected through our system will be anonymized and encrypted to prevent unauthorized access or misuse.

4.1.4 Accessibility and Inclusivity

In line with the IEEE/ACM Code of Ethics, our project aims to be accessible and inclusive, providing a virtual musical experience that can be enjoyed by a wide range of users, including those with disabilities. We will seek to design our virtual band gloves and accompanying software with user-friendly interfaces and adaptive features to accommodate diverse needs.

4.1.5 Transparency and Honest

We will be transparent and honest in our communication about the capabilities, limitations, and ongoing development of our virtual band model. This includes openly discussing any potential risks or uncertainties associated with the use of our technology, as well as our strategies for mitigating such risks.

4.1.6 Compliance with Legal and Ethical Standard

Our project will comply with all applicable laws, regulations, and ethical guidelines, including those related to human and animal testing. Should our project's development process require any form of testing that involves human participants or animals, we will obtain the necessary approvals from Institutional Review Boards (IRB) or Institutional Animal Care and Use Committees (IACUC), respectively.

4.2 Safety

Safety is paramount in any project, and the virtual band model is no exception. Various safety concerns need to be addressed to ensure the well-being of both project developers and end users:

4.2.1 Electrical Safety:

- Implement measures to prevent electrical shocks, such as using insulated cables, ensuring proper grounding, and enclosing high-voltage components.
- Clearly label areas with electrical hazards and restrict access to unauthorized personnel.
- Incorporate circuit protection devices like fuses and circuit breakers to prevent overloads.

4.2.2 Mechanical Safety:

- Ensure that all mechanical components, including moving parts and sensors, are properly designed and securely fastened to prevent accidents or injuries.
- Conduct regular inspections of mechanical components to identify and address any wear or damage that may compromise safety.

4.2.3 Lab Safety:

- Establish clear guidelines for laboratory use, including the proper handling of equipment, tools, and materials.
- Provide personal protective equipment (PPE) such as gloves and safety goggles where necessary.
- Clearly mark emergency exits and ensure that fire extinguishers and first aid kits are easily accessible.

List of Figures

References

[1] *Block Diagram for Virtual Band*. Figure 2. Page 3.
[2] *MediaPipe Hand Landmarker Demo*. Figure 10. Page 8.
[3] *MediaPipe Hand Landmarks*. Figure 9. https://developers.google.com/mediapipe/solutions/hands.
[4] *Overall CAD Diagram for Virtual Band*. Figure 3. Page 4.
[5] *Pin Layout for Raspberry PI 4B*. Figure 12. https://pinout.xyz/pinout/spi.
[6] *Preprocessing Color Space For Gloved Hands*. Figure 11. https://stackoverflow.com/questions/76325975/how-to-make-the-hand-detection-of-mediap.
[7] *Product in Context*. Figure 1. Page 2.
[8] *Raspberry Front View & Dimensions*. Figure 4. Page 4.
[9] *Raspberry Left View & Dimensions*. Figure 6. Page 5.
[10] *Raspberry Top View & Dimensions*. Figure 5. Page 5.
[11] *Schematic of TCA9548A*. Figure 16. https://cjmcu.taobao.com/.
[12] *Sensor Side View & Dimensions*. Figure 7. Page 6.