Virtual Piano Using Raspberry Pi Pico

A
Project Stage-II Report
Submitted in the Partial fulfilment of the
Academic Requirements
for the Award of the Degree of

Bachelor of Technology
in
Electronics and Communication Engineering

N. SAI SRI DHARANI	20AG1A0448
N. YASHWANTH	20AG1A0446
B. BALAJI	20AG1A0404

Under the esteemed guidance of

Mr. R. NARENDER
(Assistant Professor)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ACE
Engineering College
(NBA ACCREDITED B. TECH COURSES, ECE, EEE, CSE, MECH, CIVIL
(NAAC “A” GRADE)
An Autonomous Institution
Ankushapur (V), Ghatkesar (M), Medchal Dist.-501 301

2023-2024

ACE
ENGINEERING COLLEGE

(NBA ACCREDITED B. TECH COURSES, ECE, EEE, CSE, MECH, CIVIL)

(NAAC "A" GRADE)

An Autonomous Institution

Ankushapur (V), Ghatkesar (M), Medchal Dist.-501 301

2023-2024

CERTIFICATE

This is to certify that the project work entitled "Virtual Piano Using Raspberry Pi Pico" done by

N. SAI SRI DHARANI	20AG1A0448
N. YASHWANTH	20AG1A0446
B. BALAJI	20AG1A0404

Department of Electronics and Communication Engineering, is a record of Bonafede work carried out by them. This Project Stage-II is done as partial fulfilment of obtaining Bachelor of Technology degree to be awarded by **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD** during the academic year 2023-2024.

Mr. R. NARENDER	**DR. P. SATISH KUMAR**
(Assistant Professor)	Professor and HOD, ECE
Project Supervisor	

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to express our deepest appreciation and gratitude **Mr**. **R. NARENDER, Assistant professor**, for his invaluable guidance, constructive criticism, and encouragement during the course of this project.

Thanks to our Project Coordinators **Prof. B. GIRIRAJU** and **Mrs. A. SRITULASI**, Assistant Professor for being uniformly an excellent advisor. They were always open, helpful. and provided strong broad ideas.

We are deeply indebted to **Dr. P. SATISH KUMAR**, Head of the Department, for providing the necessary opportunities for the completion of the project.

We are grateful to **Dr. B. L. RAJU**, principal of the college, for providing an excellent atmosphere, through flexible timings to work.

We are grateful to **Prof. Y. V. GOPALA KRISHNA MURTHY**, for his moral support and for providing facilities

We are thankful to the teaching and non-teaching staff members of the electronics and communication engineering department who helped directly or indirectly for successful completion of the project.

We are very grateful to my family and friends for their constant support and encouragement during the project period.

With Regards

N. SAI SRI DHARANI 20AG1A0448

N. YASHWANTH 20AG1A0446

B. BALAJI 20AG1A0404

ABSTRACT

This project aims to construct a portable virtual piano system utilizing Raspberry Pi, touch sensors, and additional key components. Leveraging the computational prowess of Raspberry Pi and touch sensor technology, users can emulate the tactile sensation of playing a traditional piano without the constraints of physical keys. Raspberry Pi serves as the central processing unit, analysing touch sensor inputs to generate corresponding piano notes. Incorporating suggested components such as DF mini player for superior audio output, an LCD display for intuitive user interaction, a mode changer for dynamic functionality, and a robust power supply ensures an enriched musical experience. The integration of the DF mini player enhances audio quality, providing authentic sound reproduction through connected speakers or headphones. The LCD display offers real-time feedback on selected modes, settings, and musical notes being played, enhancing user engagement.

This project enables users to enjoy the advantages of portability and sound customization, empowering them to create music on the go and experiment with diverse instrument voices and sound effects. Furthermore, seamless integration with other digital music software and systems fosters collaboration and access to a wide array of music production tools.

CONTENTS

CHAPTERS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

1. Raspberry Pi Pico: RPP or **RPi Pico**
2. Secure Digital Card: **SD card**
3. Graphical User Interface: **GUI**
4. Digital Audio Workstation: **DAW**
5. Microcontroller Unit: **MCU**
6. General-Purpose Input/Output: **GPIO**
7. Capacitive Touch Sensor: **CTS**
8. Serial Peripheral Interface: **SPI**
9. Universal Asynchronous Receiver-Transmitter: **UART**
10. Liquid Crystal Display: **LCD**
11. Direct-Form Mini Player: **DF Mini Player**
12. Power Supply Unit: **PSU**
13. Amplifier: **Amp**
14. Touch Sensor Module: **TSM**
15. Operational Amplifier: **Op-Amp**

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW:

This project aims to develop a portable virtual piano system using Raspberry Pi and touch sensor technology. Leveraging Raspberry Pi's computing power and touch sensors, users can replicate the experience of playing a traditional piano without physical keys. The Raspberry Pi serves as the central processing unit, interpreting touch sensor data to generate corresponding piano notes. Key components integrated into the system include a DF mini player for high-quality audio output, an LCD display for user interaction, a mode changer for dynamic functionality switching, and a reliable power supply. These enhancements elevate the musical experience by providing superior audio quality, intuitive user interface, and seamless transitions between different functionalities.

The virtual piano offers advantages such as portability and sound customization, enabling users to create music anywhere and experiment with various instrument voices and effects. Additionally, the system facilitates easy integration with other digital music software and systems, fostering collaboration and access to diverse music production tools.

1.2 BLOCK DIAGRAM:

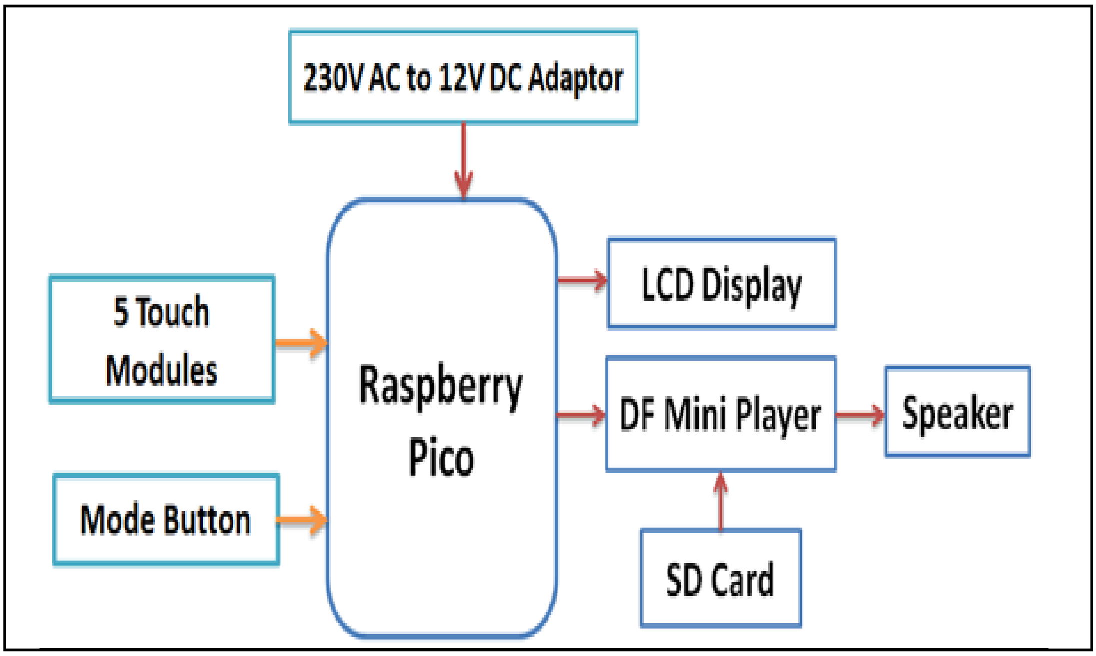

Figure 1.2 Block Diagram of Virtual Piano Using Raspberry pi Pico

1.3 <u>BLOCK DIAGRAM DESCRIPTION</u>:

1.3.1 Power Supply Unit:

The power supply unit (PSU) for your virtual piano acts like its personal energy source, ensuring that all its components receive the right amount of electricity to function properly. It regulates the electricity, keeping it stable and safe, so nothing gets too much power or draws too much energy, which could cause problems. You can connect it to different power sources, like a regular outlet or batteries, giving you flexibility depending on where you want to play. It's designed to save energy, which is great if you're using batteries. Plus, it's compact and lightweight, making it easy to carry around and connect to your piano, touch sensors, DF mini player, LCD display, mode changer, and other parts of your setup. In essence, the PSU is the unsung hero of your virtual piano, silently ensuring everything runs smoothly while you focus on creating beautiful music.

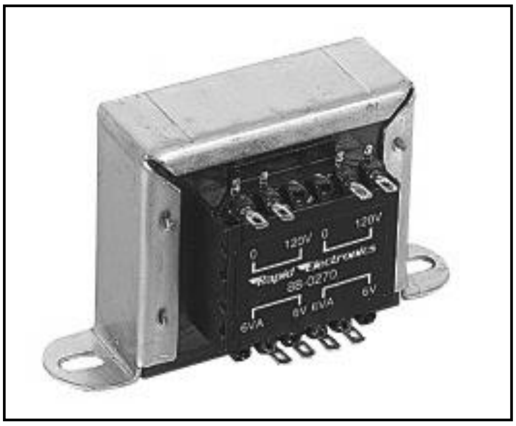

Figure 1.3.1 Step Down Transformer

1.3.2 Raspberry Pi Pico:

The Raspberry Pi Pico is a small computer chip that's great for building all sorts of electronic projects. It has a powerful brain inside called the RP2040 chip, which can do lots of different tasks quickly. The Pico has lots of pins sticking out that you can connect to other parts like sensors, lights, and buttons. It also has a USB port so you can plug it into your computer to program it and give it power. Plus, it comes with some memory built-in, so you can store programs and information on it. It's really easy to use and can be programmed in different ways, which makes it perfect for beginners and experts alike. Whether you want to build a simple toy or a complex gadget, the Raspberry Pi Pico has got you covered!

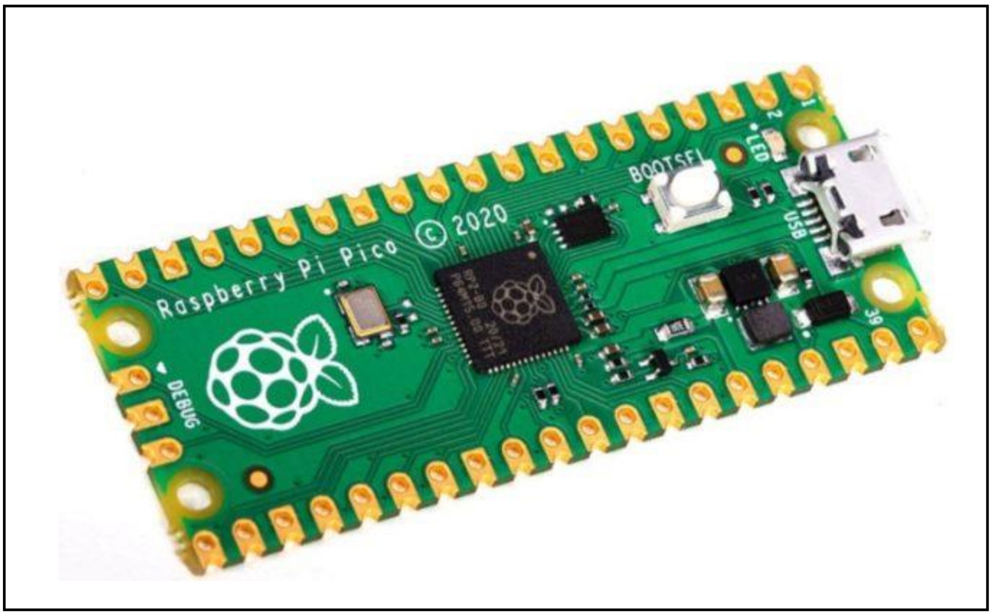

Figure 1.3.2 Raspberry Pi Pico

1.3.3 Touch Sensor:

Touch sensors are like tiny detectors that can tell when you touch them with your finger or another object. They work by sensing changes when something comes into contact with them. There are different types, but the most common ones work by either sensing changes in electricity or resistance when touched.

You've probably used touch sensors before without even realizing it! They're found in things like smartphones, tablets, and touch-sensitive buttons on appliances. They're handy because they make it easy for us to interact with electronic devices just by touching them, without needing physical buttons or switches.

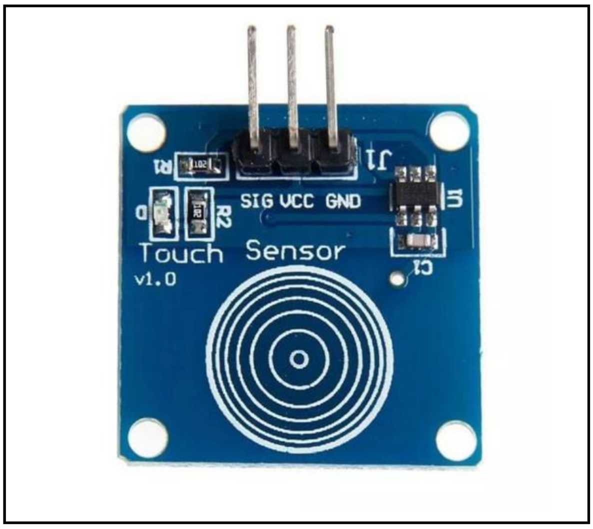

Figure 1.3.3 Touch Sensor

1.3.4 16x2 LCD Display:

A 16x2 LCD display is like a tiny screen that can show text and numbers. It's called "16x2" because it has 16 characters in each of the two rows. You've probably seen similar displays in devices like calculators, alarm clocks, or even some kitchen appliances. These displays are really handy because they can show information in a clear and easy-to-read way. They work by lighting up tiny dots to form letters, numbers, and symbols. You can use them to show things like the time, temperature, or messages. With a 16x2 LCD display, you can add a visual element to your projects, making them more interactive and informative. Plus, they're pretty simple to use and can be connected to microcontrollers like the Raspberry Pi Pico to display information from your projects.

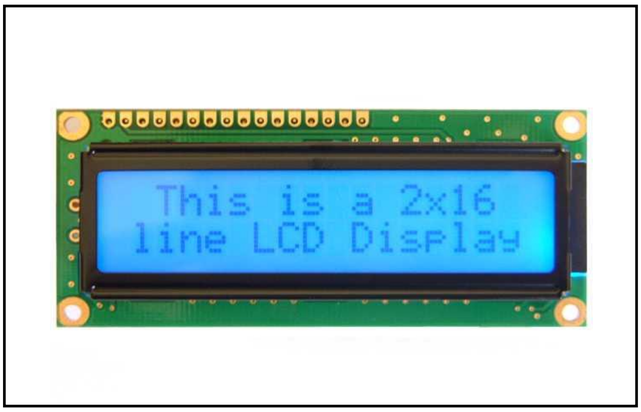

Figure 1.3.4 16x2 LCD Display

1.3.5 Speaker:

In your project, the speaker serves as the output device for the virtual piano system, producing sound based on the notes played by the user. When a user interacts with the virtual piano, the Raspberry Pi Pico processes the input from touch sensors and generates corresponding electrical signals. These signals are then converted into audio signals and sent to the speaker. The speaker is responsible for translating these electrical signals into audible sound waves, allowing the user to hear the notes they are playing. It plays a crucial role in providing feedback and enhancing the overall user experience of the virtual piano system.

Figure 1.3.5 Speaker

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

The literature survey serves as a foundational aspect of research and project development, providing insights into existing knowledge, technologies, and methodologies relevant to the project. By reviewing existing literature, researchers gain a comprehensive understanding of the field, identify gaps in knowledge, and inform their approach and design decisions.

In the context of this project, the literature survey aims to explore previous research, studies, and projects related to the development of a portable virtual piano using Raspberry Pi Pico, touch sensors, DF mini player, LCD display, mode changer, and a power supply unit. The survey will encompass various topics, including microcontroller-based musical instruments, touch sensing technologies, audio output systems, user interface design, power supply design, integration of components, and educational applications of virtual musical instruments.

Through the literature survey, key insights will be gathered regarding the capabilities and limitations of the chosen components, best practices in design and implementation, and potential challenges to anticipate. This foundational understanding will guide the development process, ensuring that the project builds upon existing knowledge and contributes to the advancement of the field.

2.2 RASPBERRY PI PICO AND MICROCONTROLLER-BASED MUSICAL INSTRUMENTS:

In recent years, people have been using small computer boards like the Raspberry Pi Pico to make musical instruments. These boards are cheap and can do lots of things, so they're great for creating synthesizers, drum machines, and other musical gadgets. Researchers have been studying how these boards work for making sounds and handling music-related tasks.

2.3 TOUCH SENSING TECHNOLOGIES IN MUSICAL INTERFACES:

In this project, touch sensing technologies, particularly capacitive touch sensing, are integral to the virtual piano interface. Capacitive touch sensors will be used on each key, enabling users to trigger notes by touching the sensor surface. Research on optimizing capacitive touch sensors for musical applications will inform sensor design to ensure reliable and responsive touch detection. Additionally, exploration of multi-touch gesture recognition and pressure sensitivity will enhance the expressive capabilities of the virtual piano. Integration with digital signal processing algorithms will enable advanced features such as polyphonic touch response and dynamic sound modulation based on touch input. Overall, touch sensing technologies play a vital role in creating an intuitive and immersive musical experience for users.

2.4 AUDIO OUTPUT SYSTEMS FOR EMBEDDED PROJECTS:

In our project, audio output systems are like speakers for our virtual piano. They're what make the piano produce sound so we can hear it. We'll explore different options for this.

DF Mini Player:

The DF Mini Player is a small device that can play audio files. It's easy to use and can play music from a memory card. We'll see if it's a good fit for our virtual piano in terms of sound quality and how well it works with the other parts.

CHAPTEP 3

DESCRIPTION

3.1 Technical Description

The virtual piano system utilizes the Raspberry Pi Pico microcontroller as its core processing unit. Capacitive touch sensors are integrated into piano keys, detecting user input. The DF Mini Player handles audio playback via serial communication with the Raspberry Pi Pico. A 16x2 LCD display provides user feedback, while a mode changer component enables mode switching. Power is supplied either via battery pack or DC adapter. Programming is done in MicroPython or C/C++, with UART for communication. GPIO pins are configured for sensor interfacing. Efficient power management and audio signal processing algorithms ensure optimal performance.

3.2 Working

- Raspberry Pi Pico initializes system components.
- Touch sensors detect key presses.
- Signal sent to Raspberry Pi Pico.
- Raspberry Pi Pico generates audio signal.
- DF Mini Player plays corresponding note.
- LCD display updates with key pressed.
- User can switch modes via mode changer.
- Virtual piano operates continuously.
- Power off Raspberry Pi Pico to stop operation.

3.3 Procedure

- Power on the Raspberry Pi Pico and initialize the system components.
- When a user touches a piano key, the corresponding capacitive touch sensor detects the touch event.
- The Raspberry Pi Pico receives the touch signal and generates the corresponding audio signal for the note to be played.
- The generated audio signal is sent to the DF Mini Player via serial communication.

- The DF Mini Player plays the pre-recorded musical note or sound associated with the received signal.
- The sound is emitted through the connected speaker.
- Simultaneously, the Raspberry Pi Pico updates the LCD display to provide visual feedback to the user.
- The display may indicate the pressed key, selected modes, settings, or other relevant information.
- If the user wishes to switch modes or settings, they can interact with the mode changer component.
- The mode changer sends a signal to the Raspberry Pi Pico to trigger the desired mode change.
- The virtual piano operates continuously, allowing users to play music, switch modes, and interact with the interface as desired.
- The virtual piano system can be terminated by powering off the Raspberry Pi Pico or disconnecting the power supply.
- All system components are reset to their initial state for future use.

3.4 Block Diagram

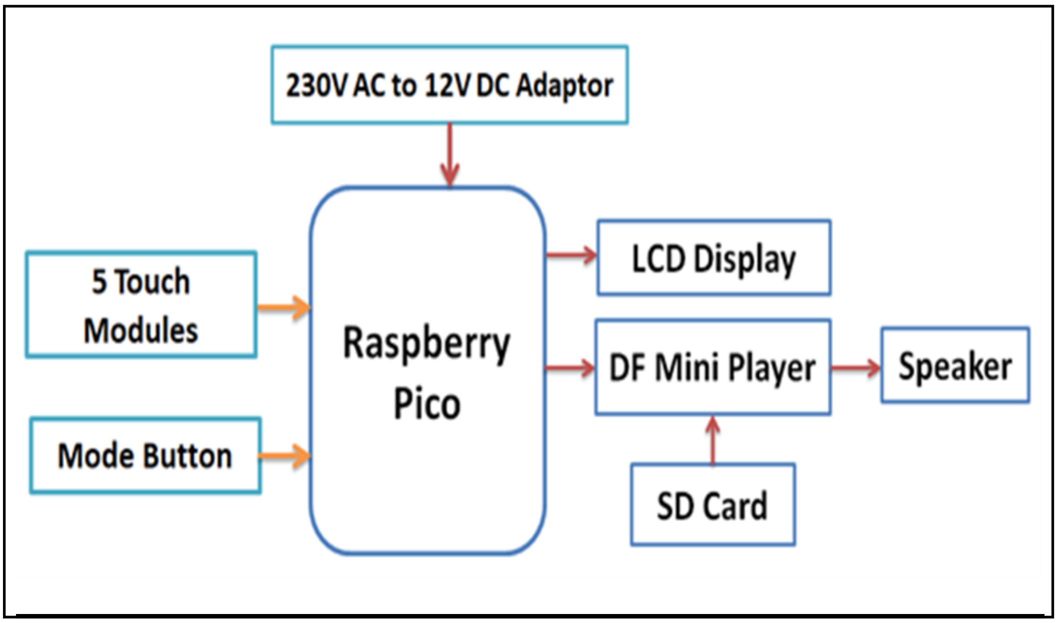

Figure 3.1.4 Block Diagram of Virtual Piano Using Raspberry pi Pico

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING MODEL:

4.1.1 Schematic Diagram

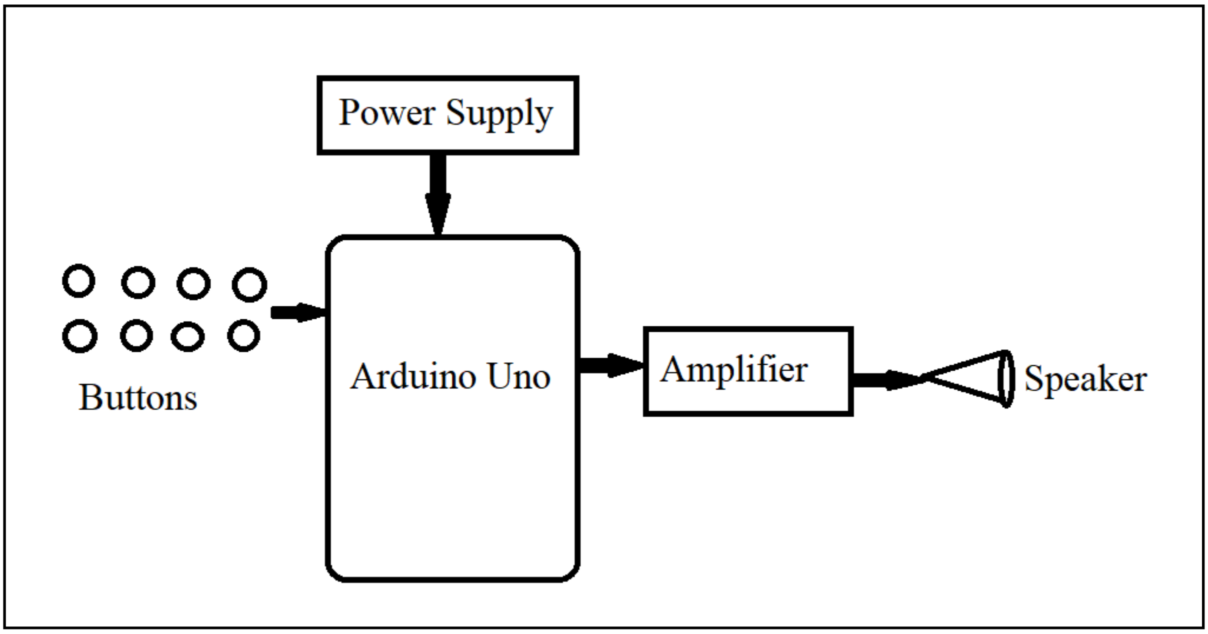

Figure 4.1.1 Schematic Diagram

The revised Schematic diagram simplifies the virtual piano system by replacing touch sensors with buttons for user interaction and the DF Mini Player with an amplifier for audio output, while eliminating the LCD display for streamlined design. In this setup, users engage with physical buttons corresponding to piano keys to trigger musical notes, with the Raspberry Pi Pico detecting button presses and generating corresponding audio signals. These signals are then amplified by the amplifier before being emitted through the speaker, resulting in audible sound. This approach offers a cost-effective alternative to the previous design, reducing complexity with fewer components while maintaining the core functionality of the virtual piano. It's important to ensure proper wiring and connections between components and to test the circuit's functionality thoroughly to verify its proper operation. Additionally, optimizing button placement for user convenience can enhance the overall usability of the system, providing an accessible and enjoyable musical experience.

4.1.2 Working

The revised virtual piano system operates through a straightforward process where users engage with physical buttons, triggering musical notes corresponding to piano keys. Upon pressing a button, a signal is sent to the Raspberry Pi Pico microcontroller, which interprets the input and generates the appropriate audio signal for the desired note. This audio signal is then routed to an amplifier, which increases its strength before transmitting it to a connected speaker. As a result, the user hears the produced sound through the speaker, completing the interactive experience. This simplified setup offers an accessible and intuitive interface for users to engage with the virtual piano, enabling them to play music effectively using a tactile button interface. Through continuous operation, users can enjoy exploring musical compositions and melodies, making the virtual piano system an engaging and enjoyable platform for musical expression.

4.1.3 Advantages and Disadvantages

Advantages

- Cost-Effective
- Customization
- Accessibility
- Community Support and Documentation

Disadvantage

- Dependency on Power Supply
- Low Quality Audio

4.2 PROPOSED MODEL

4.2.1 Circuit Diagram

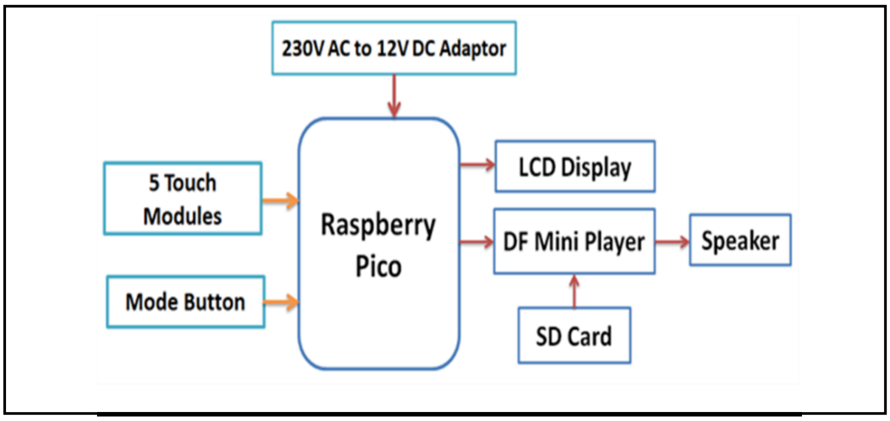

Figure 4.2.1 Circuit Diagram

The virtual piano system utilizes the Raspberry Pi Pico microcontroller as its core processing unit. Capacitive touch sensors are integrated into piano keys, detecting user input. The DF Mini Player handles audio playback via serial communication with the Raspberry Pi Pico. A 16x2 LCD display provides user feedback, while a mode changer component enables mode switching. Power is supplied either via battery pack or DC adapter. Programming is done in MicroPython or C/C++, with UART for communication. GPIO pins are configured for sensor interfacing. Efficient power management and audio signal processing algorithms ensure optimal performance.

4.2.2 Working

- Raspberry Pi Pico initializes system components.
- Touch sensors detect key presses.
- Signal sent to Raspberry Pi Pico.
- Raspberry Pi Pico generates audio signal.
- DF Mini Player plays corresponding note.
- LCD display updates with key pressed.
- User can switch modes via mode changer.
- Virtual piano operates continuously.
- Power off Raspberry Pi Pico to stop operation.

CHAPTER 5

THEORETICAL ANALYSIS

5.1 TECHNICAL DESCRIPTION:

The portable virtual piano system integrates Raspberry Pi technology with touch sensor input, advanced audio output via the DF Mini Player, and user-friendly interaction features such as an LCD display and mode changer. Users trigger notes by touching capacitive sensors, with the Raspberry Pi processing these inputs to generate corresponding audio signals. The DF Mini Player reproduces these signals through connected speakers, ensuring authentic sound reproduction. An LCD display provides visual feedback on selected modes and notes being played, enhancing user engagement. With its compact design and seamless integration with digital music software, the system offers users a versatile and immersive musical experience, empowering them to create music anywhere, anytime.

The main blocks of this project are:

1. Power Supply
2. Raspberry Pico
3. Touch Sensor
4. Push Button
5. 16x2 LCD Display
6. DF Mini Player
7. Speaker
8. SD Card

5.1.1 Power Supply

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A RPS (Regulated Power Supply) is the Power Supply with Rectification, Filtering and Regulation being done on the AC mains to get a Regulated power supply for Microcontroller and for the other devices being interfaced to it.

A power supply unit can by broken down into a series of blocks, each of which performs a particular function. A DC power supply which maintains the output voltage constant irrespective of AC mains fluctuations or load variations is known as “Regulated D.C Power Supply”.

For example, a 5V regulated power supply system as shown below:

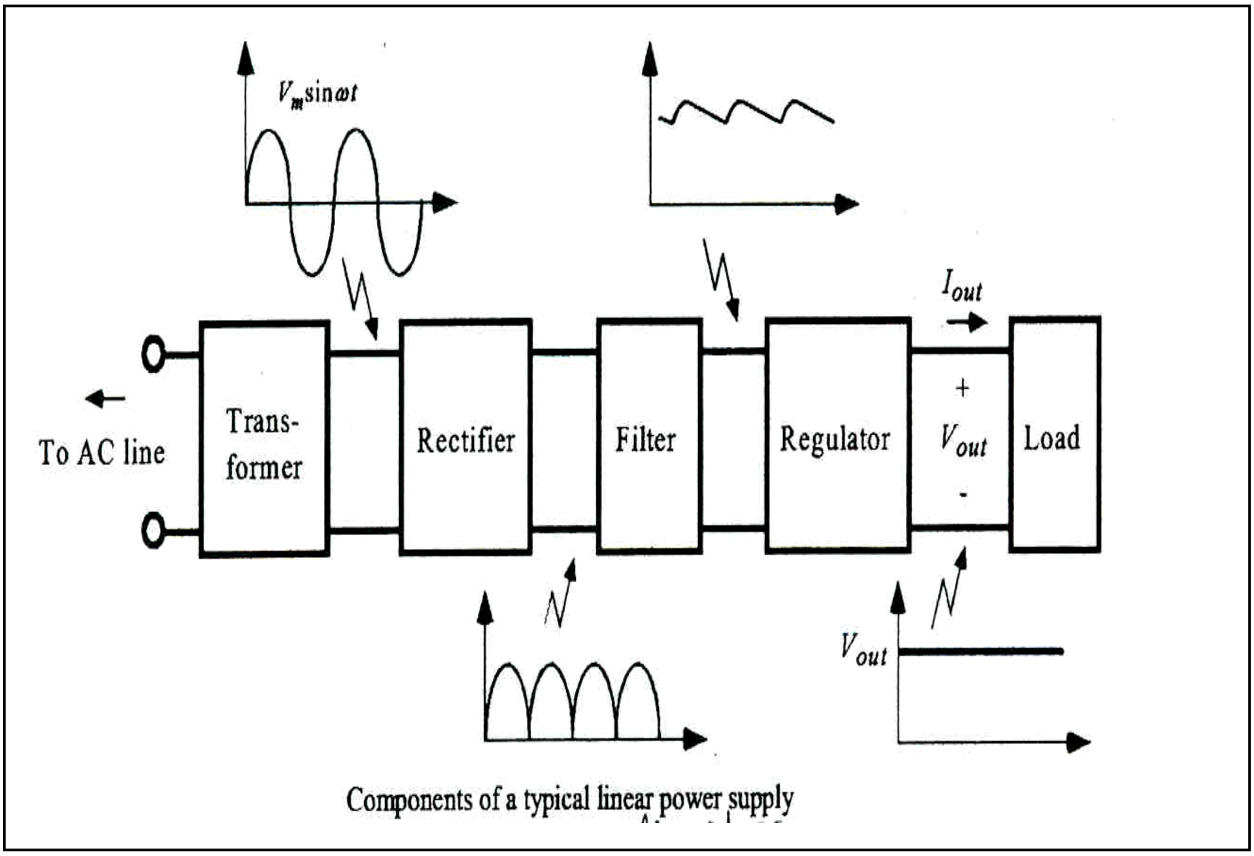

Figure 5.1.1 Block Diagram of Power Supply

1. Transformer
2. Rectifier
3. Filter
4. Regulator

5.1.1.1 TRANSFORMER

A transformer is an electrical device which is used to convert electrical power from one Electrical circuit to another without change in frequency. Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC. Step-up transformers increase in output voltage, step-down transformers decrease in output voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage to a safer low voltage.

The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead, they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core.   Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up. The ratio of the number of turns on each coil, called the turn's ratio, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

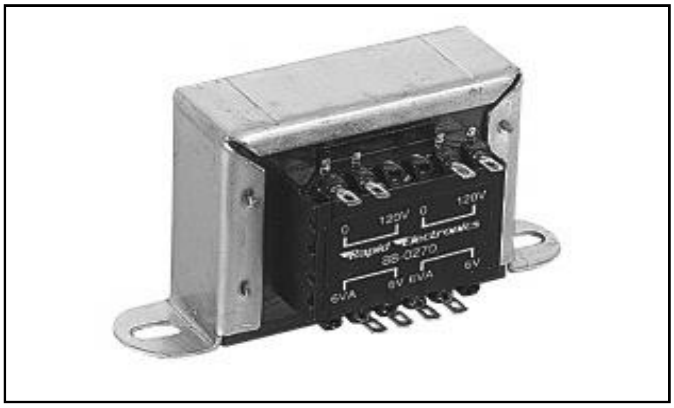

Figure 5.1.1.1 Transformer

Turns ratio = $V_p / V_S = N_p / N_S$

Power Out= Power In

$V_S \times I_S = V_P \times I_P$

V_p = primary (input) voltage

N_p = number of turns on primary coil

I_p = primary (input) current

5.1.1.2 RECTIFIER

A circuit which is used to convert a.c to dc is known as RECTIFIER. The process of conversion a.c to d.c is called "rectification".

Types of Rectifiers

1. Half wave Rectifier
2. Full wave Rectifier

(i). Centre tap full wave rectifier.

(ii). Bridge type full bridge rectifier.

Parameter	Type of Rectifier		
	Half wave	**Full wave**	**Bridge**
Number of diodes	1	2	4
PIV of diodes	Vm	2Vm	Vm
D.C output voltage	V_m/π	$2V_m/\pi$	$2V_m/\pi$
Vdc, at no-load	0.318Vm	0.636Vm	0.636Vm
Ripple factor	1.21	0.482	0.482
Ripple frequency	f	2f	2f
Rectification efficiency	0.406	0.812	0.812
Transformer Utilization Factor (TUF)	0.287	0.693	0.812
RMS voltage Vrms	Vm/2	Vm/√2	Vm/√2

Table 5.1.1.2: Comparison of Rectifier Circuits

Full-wave Rectifier

From the above comparison we came to know that full wave bridge rectifier as more advantages than the other two rectifiers. So, in our project we are using full wave rectifier circuit.

5.1.1.3 FILTER

A Filter is a device which removes the a.c component of rectifier output but allows the d.c component to reach the load

Capacitor Filter

We have seen that the ripple content in the rectified output of half wave rectifier is **121%** or that of full-wave or bridge rectifier or bridge rectifier is **48%** such high percentages of ripples is not acceptable for most of the applications.

Ripples can be removed by one of the following methods of filtering.

(a) A capacitor, in parallel to the load, provides an easier by –pass for the ripples voltage though it due to low impedance. At ripple frequency and leave the D.C. to appear at the load.

(b) An inductor, in series with the load, prevents the passage of the ripple current (due to high impedance at ripple frequency) while allowing the d.c (due to low resistance to d.c)

(c) Various combinations of capacitor and inductor, such as L-section filter section filter, multiple section filter etc. which make use of both the properties mentioned in (a) and (b) above. Two cases of capacitor filter, one applied on half wave rectifier and another with full wave rectifier.

Filtering is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output. Filtering significantly increases the average DC voltage to almost the peak value (1.4 × RMS value).

To calculate the value of capacitor(C),

$$C = \frac{1}{4} * \sqrt{3} * f * r * Rl$$

Where

f = supply frequency,

r = ripple factor,

Rl = load resistance

Note: In our circuit we are using 1000µF hence large value of capacitor is placed to reduce ripples and to improve the DC component.

5.1.1.4 REGULATOR:

Voltage regulator ICs is available with fixed (typically 5, 12 and 15V) or variable output voltages. The maximum current they can pass also rates them. Negative voltage regulators are available, mainly for use in dual supplies.

Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection'). Many of the fixed voltage regulators ICs have 3 leads and look like power transistors, such as the 7805 +5V 1A regulator shown on the right. The LM7805 is simple to use. You simply connect the positive lead of your unregulated

DC power supply (anything from 9VDC to 24VDC) to the Input pin, connect the negative lead to the Common pin and then when you turn on the power, you get a 5 volt supply from the output pin.

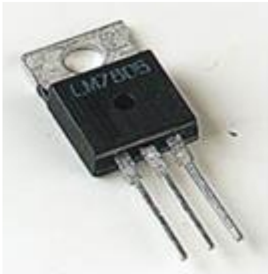

Figure 5.1.1.4 Three Terminal Voltage Regulator

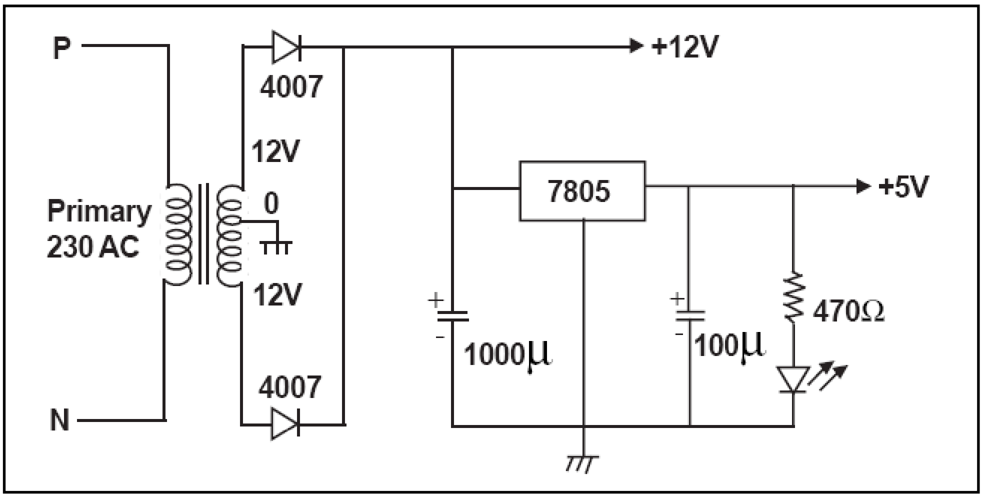

Figure 5.1.1 Power Supply

5.1.2 RASPBERRY PI PICO

In this Getting Started tutorial, we will learn about the **Raspberry Pi Pico**, a brand-new exciting Microcontroller board based on **RP2040 Microcontroller** from the **Raspberry Pi Foundation**. The Raspberry Pi Pico is a low-cost Arm-based microcontroller that we can program using **C/C++** and **MicroPython**. Over the years Raspberry Pi boards have become a must tool for students, hobbyists or Industrialists. But when it comes to cost, the Raspberry Pi Board is overtaken by **Arduino, ESP32, STM32**, or other **AVR, ARM, PIC** Microcontrollers. The Raspberry Pi computer costs around **35-40$** whereas the other microcontrollers barely cost **2-5$** only. This is the reason why **Raspberry Pi Foundation** released their low-cost powerful competitive Raspberry Pi Pico Board with RP2040, a **Dual Core ARM Cortex-M0+ Microcontroller**.

The tutorial covers the RP2040 Microcontroller, its **features** & **specifications**. We will also learn about the Raspberry Pi Pico Board, its layout, and specifications. The detailed guide of Raspberry Pi Pico Pins like **ADC pins, I2C Pins, SPI Pins, UART**, etc can help you to interface any **sensors** or **module** with this powerful board.

Since it's a **Raspberry Pi Pico getting started tutorial**, so we will only program the device using Micropython. For that, you can either use **Thonny IDE** or you can also go with **uPyCraft IDE**. In some other tutorial, we will learn how to program Raspberry Pi Pico with C/C++. Even the **Arduino IDE** will support Raspberry Pie Pico in future as it is under development phase now. We will go through the basic **Raspberry Pi Pico LED Blink Code** & check the board functionality.

5.1.2.1 WHAT IS RP2040 MICROCONTROLLER?

Earlier all the Raspberry Pi boards like Raspberry Pi 3 **or** 4 **or** Raspberry Pi Zero **featured Broadcom Processors like** BCM2835, BCM2836, BCM2711 **etc. The RP2040 chip was announced on 21st January 2021 and is the first processor designed by the Raspberry Pi Foundation.**

Figure 5.1.2.1 RP2040 Microcontroller

The RP2040 is a **32-bit** dual **ARM Cortex-M0+ microcontroller** integrated circuit released at the same time as part of the Raspberry Pi Pico board. The processor is a low-cost microcontroller and costs around **US$4**. The chip is **40nm** silicon in a **7×7 mm QFN-56** package.

The RP2040 contains two **ARM Cortex-M0+** cores clocked at **133 MHz** together with **264 KB** of RAM. The Program memory is external and supports up to **16 MB**. The device has everything you expect from a modern microcontroller like **UARTS, SPI,** and **I2C** ports, and there are timers, PWM, DMA, and a 12-bit analog-to-digital converter **(ADC)**.

5.1.2.2 MEANING OF RP2040:

- The name RP2040 has a very interesting meaning explained below.
- **RP means**: Raspberry Pi
- **Number 2 means**: Processor Cores as it is a dual-core microcontroller. So, the value is 2**.**
- **Number 0 means**: Type of Processor Core as it is ARM Cortex-M0+. So, the value is 0.
- **Number 4 means**: Represents On-chip RAM. RP2040 has 264 KB of RAM. The formula to get 4 value is: floor (log2 (ram / 16k)). So, the value is 4.
- **Number 0 means**: Represents On-chip Flash. As there is no on-chip flash, the value is 0.

5.1.2.3 RP2040 KEY FEATURES:

1. 133MHz dual ARM Cortex-M0+ cores
2. 264kB SRAM in six independent bank
3. Support for up to 16MB of off-chip Flash memory via dedicated QSPI bus
4. DMA controller
5. Fully-connected AHB crossbar
6. Interpolator and integer divider peripherals
7. On-chip programmable LDO (Low-dropout regulator) to generate core voltage
8. 2 on-chip PLLs to generate USB and core clocks
9. 30 GPIO pins, 4 of which can be used as analog inputs

5.1.2.4 INTRODUCTION TO RASPBERRY PI PICO:

The Raspberry Pi Pico is the first microcontroller board based on the RP2040. It looks a lot like other **microcontroller boards** with the MCU in the center, a micro-USB connector on one end, and a row of contacts along each side. A 3-pin **debug connector** is available at the other end of the board.

The Raspberry Pi Pico measures **51 by 21 mm**, which is the exact same size as an ESP32 Pico Kit & slightly larger than an Arduino Nano or Micro.

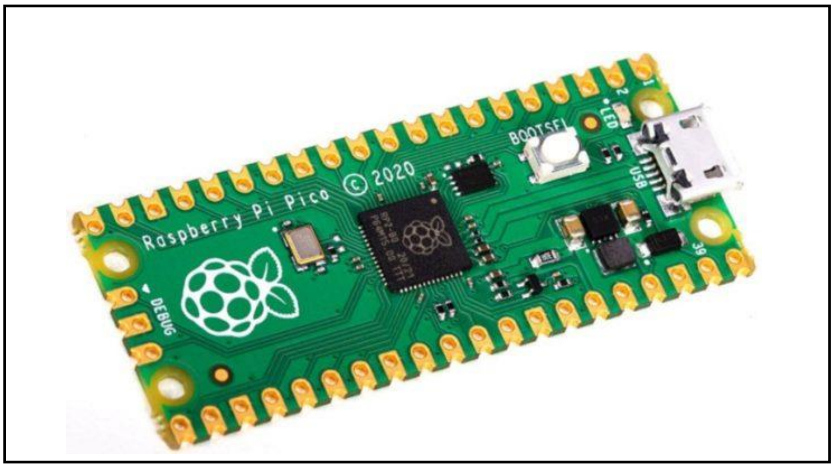

Figure 5.1.2.4(1) Raspberry Pi Pico

The Pico comes with **2 MB of QSPI Flash memory** and 25 of the 30 GPIO pins of the RP2040 have been brought out on the extension connectors. The board is **breadboard friendly** and fits perfectly on breadboard.

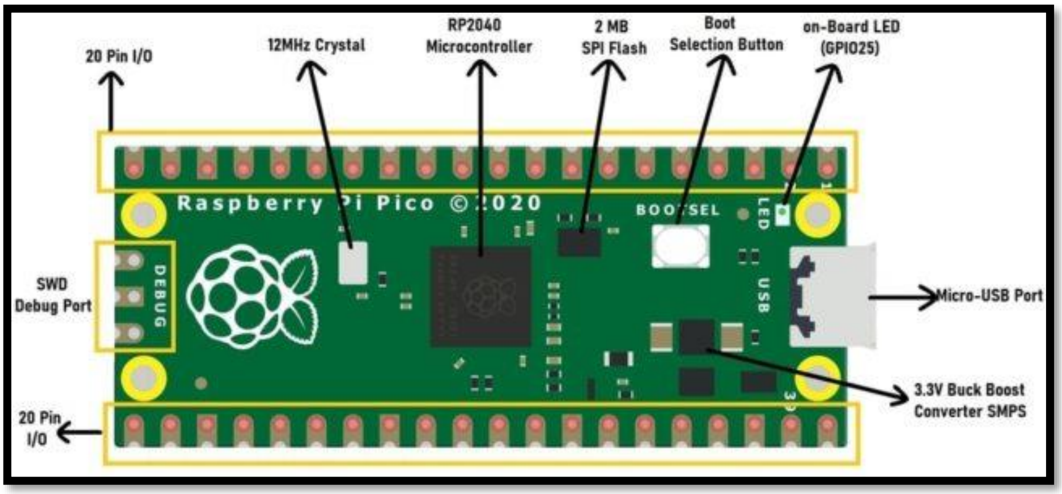

Figure 5.1.2.4(2) Raspberry Pi Pico

5.1.2.5 FEATURES OF RASPBERRY PI PICO:

Following are the features of Raspberry Pi Pico Board.

1. Based on **RP2040** Microcontroller
2. **2 MB** of SPI Flash Memory
3. **Type B Micro-USB** port for power & programming
4. **40 DIP** style IO Pins
5. 3-pin ARM Serial Wire Debug (SWD) interface
6. **12 MHz Crystal** oscillator
7. Boot Selection Button

8. Programmable LED connected to GPIO 25
9. **3.3V** Fixed Output **Buck-Boost SMPS Converter**

5.1.2.6 RASPBERRY PI PICO PINOUT:

This is a top view of the pinouts on the Raspberry Pi Pico. The pin labels are on the bottom of the board.

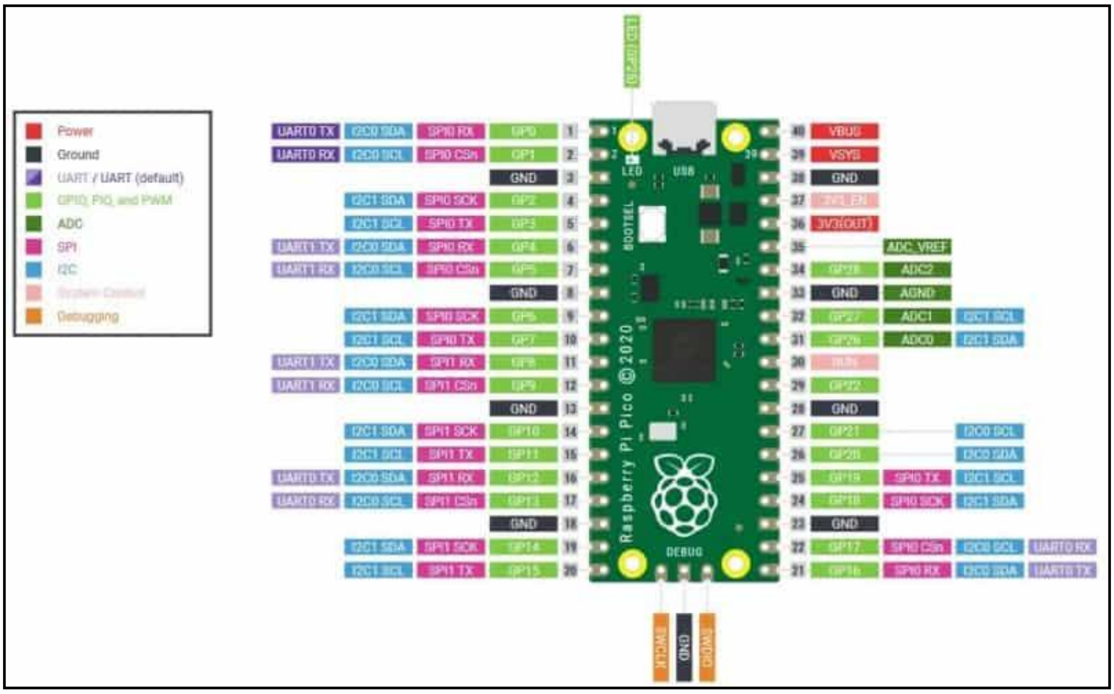

Figure 5.1.2.6 Raspberry Pi Pico Pinout

There is **40-pin** on the Raspberry Pi Pico. Out of those **40 pins**, 26 pins are Input-Output (IO Pins). All those 14 pins are analog, digital, and other Serial Pins. There are **14 power** and system-related pins. The remaining 3 more pins are used for **SWD Debugging**.

There are two I2C peripherals available, I2C0 and I2C1. Similarly, there are two SPI peripherals; SPI0 and SPI1The number of UART Pins are also two, UART0 and UART1. You can assign any of these to the pins on which they are available.

Before you start using Raspberry Pi Pico, you have to **solder 40 pin male headers**, 20 on each side of the board.

5.1.3 TOUCH SENSOR

Touch sensors are like tiny detectors that can tell when you touch them with your finger or another object. They work by sensing changes when something comes into contact with them. There are different types, but the most common ones work by either sensing changes in electricity or resistance when touched.

You've probably used touch sensors before without even realizing it! They're found in things like smartphones, tablets, and touch-sensitive buttons on appliances. They're handy because they make it easy for us to interact with electronic devices just by touching them, without needing physical buttons or switches.

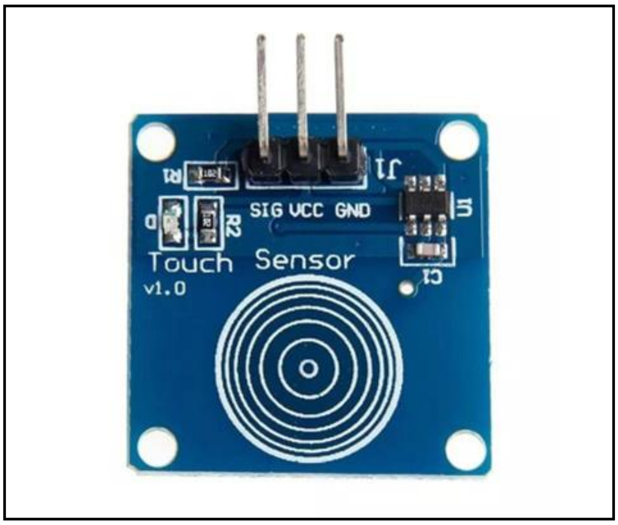

Figure 5.1.3 Touch Sensor

5.1.3.1 PRINCIPLE OF OPERATION:

Capacitive touch sensors operate on the fundamental principle of capacitance, which refers to the ability of an electrical system to store charge. In the context of touch sensors, capacitance is affected by the proximity or touch of a conductive object, such as a finger. When a user's finger approaches or makes contact with the surface of the sensor, it alters the capacitance of the sensor's electrodes. This change in capacitance is then detected by the sensor's circuitry, which interprets it as a touch event. By measuring variations in capacitance, capacitive touch sensors can accurately detect touch input without the need for physical pressure, making them ideal for applications where light touch or proximity-based interaction is desired.

5.1.3.2 ELECTRODE STRUCTURE:

Typically, capacitive touch sensors consist of electrodes arranged in a specific pattern on a substrate material, such as glass or plastic. These electrodes are connected to the sensor's circuitry and form a capacitive sensing field that extends outward from the sensor's surface.

When a conductive object, such as a user's finger, enters this sensing field, it alters the capacitance between the electrodes. The size, shape, and arrangement of the electrodes can vary depending on the specific application and desired sensitivity of the sensor. By strategically designing the electrode layout, manufacturers can optimize the sensor's performance for different use cases, such as touchscreens, proximity sensors, or gesture recognition systems.

5.1.3.4 SIGNAL PROCESSING:

Once a touch event occurs, the sensor's circuitry measures the change in capacitance and converts it into an electrical signal representing the touch input. This signal is then transmitted to a microcontroller or other processing unit for further analysis and interpretation. In the context of your project, the Raspberry Pi Pico serves as the central processing unit responsible for receiving and processing signals from the touch sensors. By analyzing the touch input, the Raspberry Pi Pico can generate corresponding musical notes or trigger other actions within the virtual piano system. This seamless integration of touch sensors with the microcontroller enables intuitive and responsive user interaction, enhancing the overall user experience of the virtual piano.

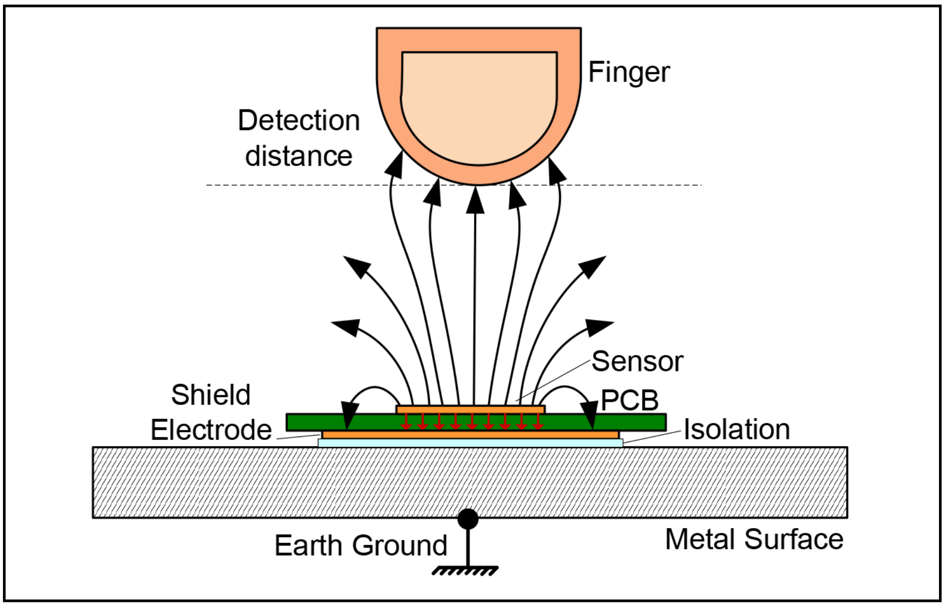

Figure 5.1.3.4 Signal Processing of Touch Sensor

5.1.4 PUSH BUTTON

Push buttons, also known as momentary switches or tactile switches, are common input devices used to provide manual control or user interaction in electronic systems. They consist of a button that, when pressed, makes or breaks electrical contact momentarily, triggering an action or event in the system. Here's an overview of push buttons:

Figure 5.1.4 Push Switch

5.1.4.1 STRUCTURE AND OPERATION:

Push buttons typically consist of a button cap, housing, and terminals. The button cap is the part that users press, while the housing encloses the internal components. Inside the housing, there are usually one or more metal contacts and a spring mechanism. When the button is pressed, the contacts come into contact with each other, completing the electrical circuit temporarily. When the button is released, the contacts separate again, breaking the circuit.

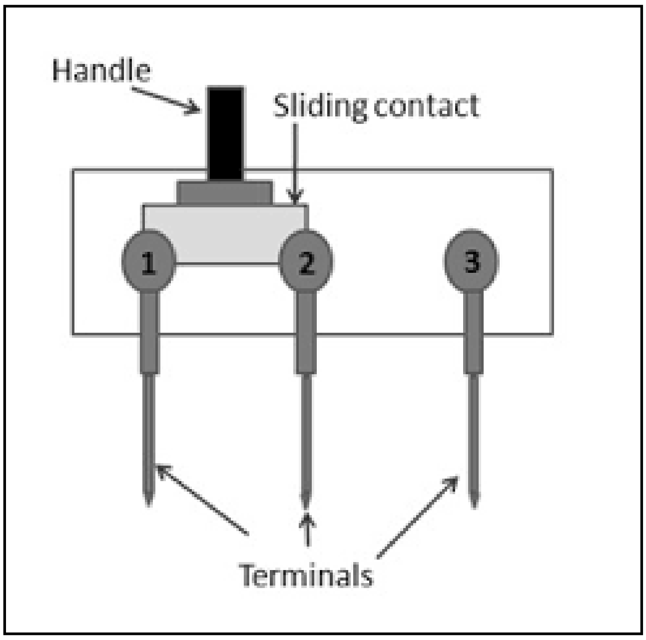

Figure 5.1.4.1 Structure of Slide Switch

5.1.4.2 APPLICATIONS:

Push buttons are used in a wide range of electronic devices and systems, including appliances, industrial control panels, consumer electronics, and automotive controls. They provide a simple and reliable means of user input, allowing users to initiate actions or control functions with ease. In your project, push buttons can serve as an alternative to capacitive touch sensors for triggering musical notes in the virtual piano system.

Users can press the buttons corresponding to piano keys to play notes, providing a tactile and intuitive interface for musical interaction.

5.1.5 LIQUID CRYSTAL DISPLAY:

5.1.5.1 WHAT IS THE LCD 16×2?

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments.

The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

An LCD screen is an electronic display module that uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates o a display 16 characters per line in 2 such lines.

In this LCD each character is displayed in a 5×7 pixel matrix. Character LCD's come in many sizes 8x1, 8x2, 10x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 30x2, 32x2, 40x2 etc. Many multinational companies like Philips, Hitachi, and Panasonic make their own custom type of character LCD's to be used in their products.

All character LCD's performs the same functions (display characters numbers special characters, asci characters etc.).Their programming is also same and they all have same 14 pins (0-13) or 16 pins (0 to 15).

The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

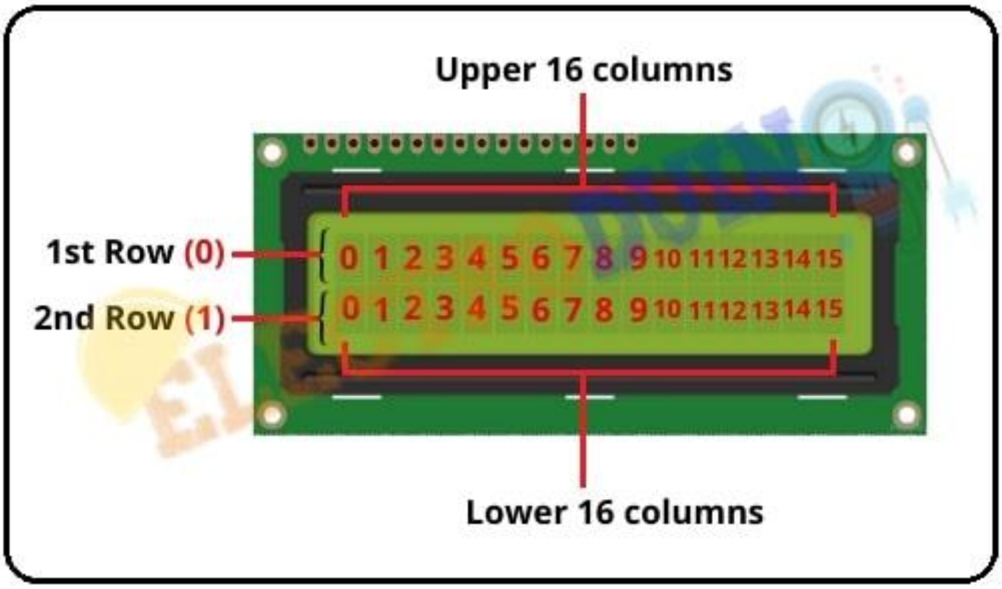

Figure 5.1.5.1 LCD Display

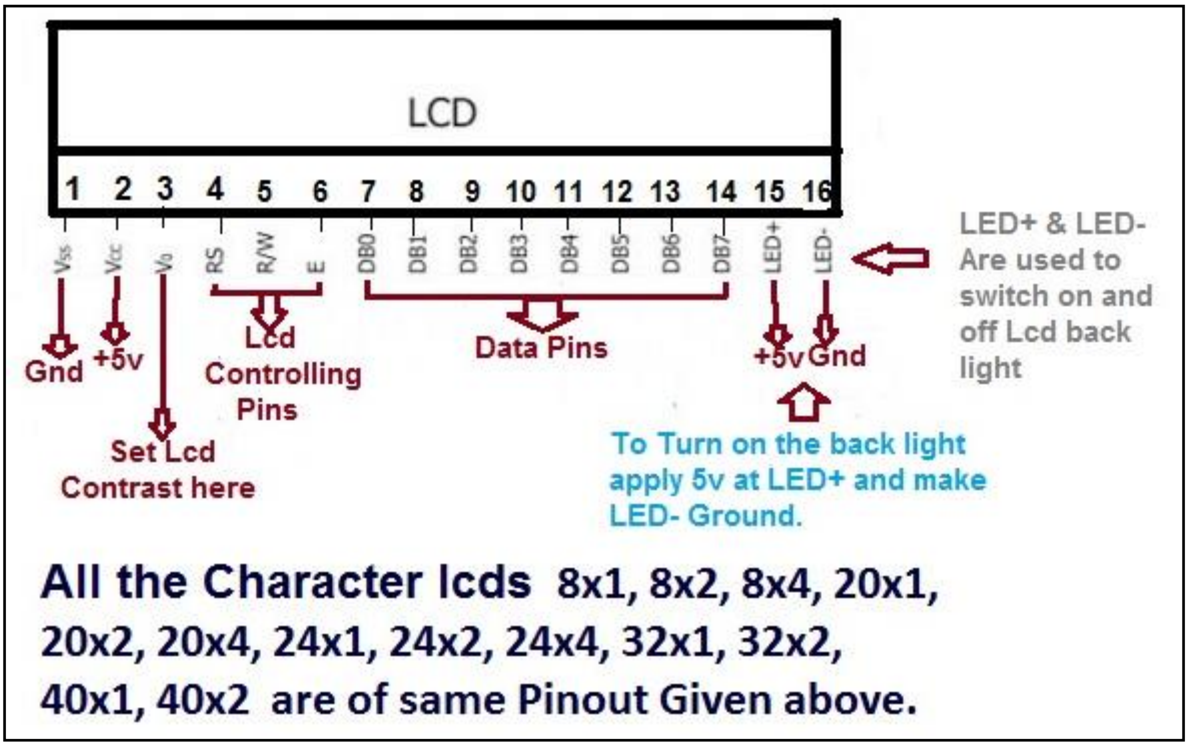

Figure 5.1.5.1(2) 16x2 LCD Display

5.1.5.2 LCD PINOUT DESCRIPTION:

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.

- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

5.1.5.3 FEATURES OF LCD16X2

The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8-pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight
- It displays a few custom generated characters

5.1.5.4 WORKING PRINCIPLE

The basic **working principle of LCD** is passing the light from layer to layer through modules. These modules will vibrate & line up their position on 90^{o} that permits the polarized sheet to allow the light to pass through it. These molecules are accountable for viewing the data on every pixel. Every pixel utilizes the method of absorbing light to illustrate the digit. To display the value, the position of molecules must be changed to the angle of light.

So this light deflection will make the human eye notice the data that will be the ingredient wherever the light gets absorbed. Here, this data will supply to the molecules & will be there till they get changed. At present, LCDs are used frequently in CD/DVD players, digital watches, computers, etc. In screen industries, LCDs have replaced the CRTs (Cathode Ray Tubes) because these displays use more power as compared to LCD, heavier & larger.

The displays of LCDs are thinner as compared to CRTs. As compared to LED screens, LCD has less power consumption because it functions on the fundamental principle of blocking light instead of dissipating.

Registers of LCD

A 16×2 LCD has two registers like data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is ‘0’, then it is known as command register. Similarly, when the register set is ‘1’, then it is known as data register.

Command Register

The main function of the command register is to store the instructions of command which are given to the display. So that predefined tasks can be performed such as clearing the display, initializing, set the cursor place, and display control. Here commands processing can occur within the register.

Data Register

The main function of the data register is to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

16×2 LCD Commands

The commands of LCD 16X2 include the following.

- For Hex Code-01, the LCD command will be the clear LCD screen
- For Hex Code-02, the LCD command will be returning home
- For Hex Code-04, the LCD command will be decrement cursor
- For Hex Code-06, the LCD command will be Increment cursor
- For Hex Code-05, the LCD command will be Shift display right
- For Hex Code-07, the LCD command will be Shift display left
- For Hex Code-08, the LCD command will be Display off, cursor off
- For Hex Code-0A, the LCD command will be cursor on and display off
- For Hex Code-0C, the LCD command will be cursor off, display on
- For Hex Code-0E, the LCD command will be cursor blinking, Display on
- For Hex Code-0F, the LCD command will be cursor blinking, Display on
- For Hex Code-10, the LCD command will be Shift cursor position to left
- For Hex Code-14, the LCD command will be Shift cursor position to the right
- For Hex Code-18, the LCD command will be Shift the entire display to the left
- For Hex Code-1C, the LCD command will be Shift the entire display to the right
- For Hex Code-80, the LCD command will be Force cursor to the beginning (1st line)
- For Hex Code-C0, the LCD command will be Force cursor to the beginning (2nd line)
- For Hex Code-38, the LCD command will be 2 lines and 5×7 matrix
- For Hex Code-0F, the LCD command will be cursor blinking, Display on

5.1.6 DF MINI PLAYER:

The DF Player Mini is a small and low-cost MP3 module player with a simplified output directly to the speaker. The DF Player mini standalone can be used as a stand-alone module with an attached battery, speaker, and push buttons or used in combination with an Arduino UNO or any other with RX/TX capabilities.

This DF robot mp3 player perfectly integrates the hard decoding module, which supports common audio formats such as MP3, WAV, and WMA. Besides, the DF player mini mp3 player module also supports TF card with FAT16, FAT32 file system. Through a simple serial port, you can play the designated music without any other tedious underlying operations.

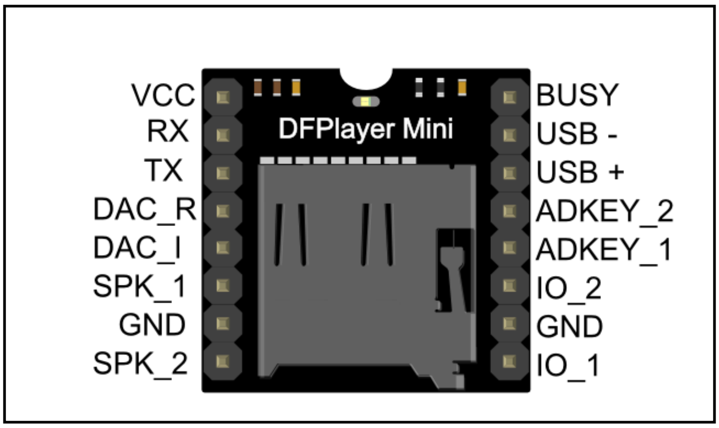

Figure 5.1.6 DF Player mini module.

5.1.6.1 APPLICATIONS OF DF MINI PLAYER:

- Car navigation voice broadcast;
- Road transport inspectors, toll stations voice prompts;
- Railway station, bus safety inspection voice prompts;
- Electricity, communications, financial business hall voice prompts;
- Vehicle into and out of the channel verify that the voice prompts;
- The public security border control channel voice prompts;
- Multi-channel voice alarm or equipment operating guide voice;
- The electric tourist car safe driving voice notices;
- Electro mechanical equipment failure alarm;
- Fire alarm voice prompts;
- The automatic broadcast equipment, regular broadcast.

5.1.6.2 SPECIFICATIONS:

- Supported sampling rates (kHz): 8/11.025/12/16/22.05/24/32/44.1/48
- 24 -bit DAC output, support for dynamic range 90dB, SNR support 85dB
- Fully supports FAT16, FAT32 file system, maximum support 32G of the TF card, support 32G of U disk, 64M bytes NOR FLASH
- A variety of control modes, I/O control mode, serial mode, AD button control mode
- Advertising sound waiting function, the music can be suspended. when advertising is over in the music continue to play
- Audio data sorted by folder supports up to 100 folders, every folder can hold up to 255 songs

- 30 level adjustable volume, 6 -level EQ adjustable

5.1.7 SPEAKER

A speaker is an essential component in audio output systems, converting electrical signals into sound waves that can be heard by humans. In your virtual piano project, a speaker plays a crucial role in producing the musical notes generated by the Raspberry Pi Pico or any other audio source. Here's an overview of speakers and their use in your project:

Figure 5.1.7 Speaker

5.1.7.1 STRUCTURE AND OPERATION:

A speaker typically consists of several key components, including a diaphragm, voice coil, magnet, and enclosure. When an electrical signal is applied to the speaker's voice coil, it creates a magnetic field that interacts with the permanent magnet, causing the voice coil and attached diaphragm to move back and forth. This movement generates sound waves by compressing and rarefying the air in front of the speaker, producing audible sound.

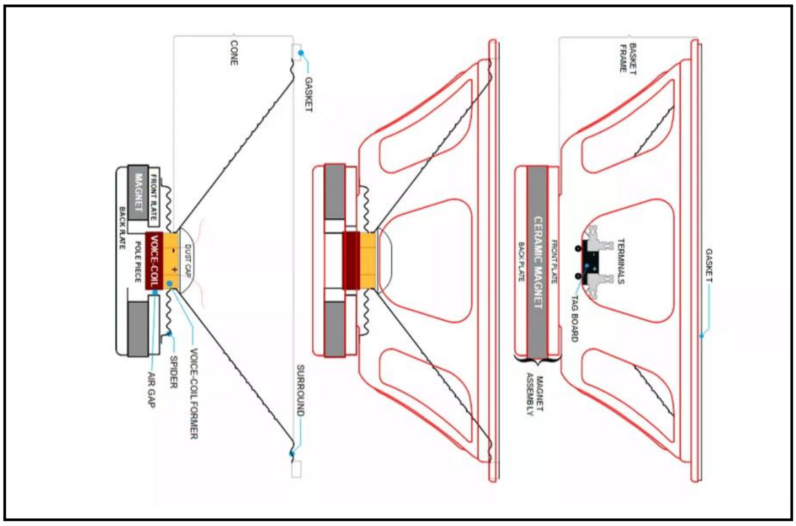

Figure 5.1.7.1 Structure of Speaker

5.1.7.2 TYPES OF SPEAKERS:

There are various types of speakers available, each with its own characteristics and applications. Common types include:

Dynamic Speakers: These are the most common type of speaker, known for their versatility and wide range of applications.

Piezoelectric Speakers: These speakers use a piezoelectric crystal to produce sound waves and are often used in applications where space is limited or power efficiency is critical.

Electrostatic Speakers: These speakers use an electrically charged diaphragm suspended between two perforated metal plates to produce sound and are known for their high-fidelity audio reproduction.

5.1.7.3 AUDIO OUTPUT IN YOUR PROJECT:

In your virtual piano project, the speaker is responsible for reproducing the musical notes generated by the Raspberry Pi Pico or any other audio source. The Raspberry Pi Pico generates electrical signals representing the musical notes based on user input from touch sensors or push buttons. These signals are then sent to the speaker, which converts them into sound waves that can be heard by the user.

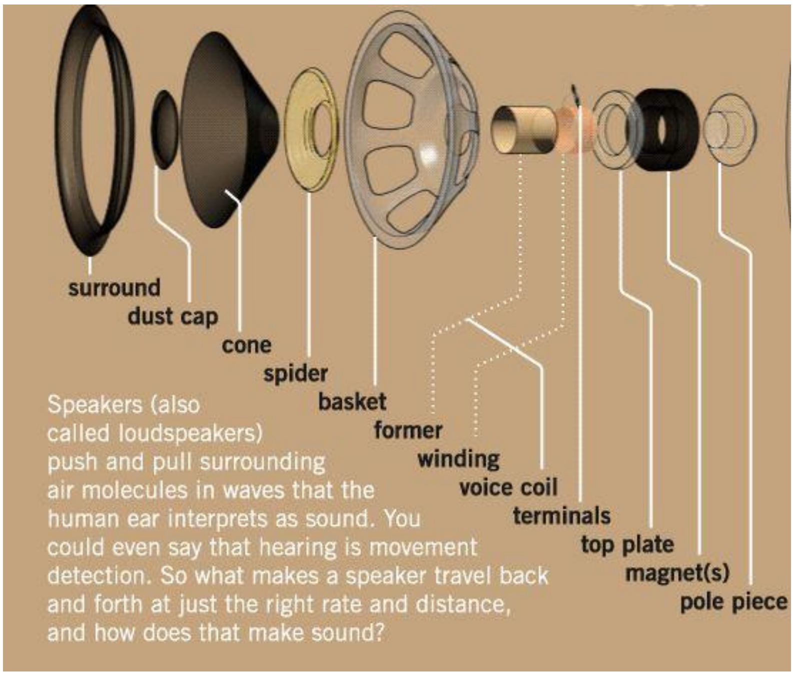

Figure 5.1.7.4 Audio Output

5.1.8 SD CARD

An SD (Secure Digital) card, commonly used in portable electronic devices like the Raspberry Pi, is a compact flash memory storage medium available in various physical sizes and capacities. In your virtual piano project, the SD card serves as the primary storage for the Raspberry Pi's operating system and user data, enabling the device to boot and run applications. By downloading and writing the appropriate operating system image to the SD card, users can configure their Raspberry Pi to function as a virtual piano system, storing additional files such as audio samples and MIDI files for enhanced musical capabilities.

When selecting an SD card for the project, considerations such as storage capacity, speed class, and brand reputation are important to ensure optimal performance and reliability. Overall, the SD card plays a crucial role in facilitating the operation and expansion of the Raspberry Pi-based virtual piano, providing a versatile and flexible storage solution for users to create and enjoy music

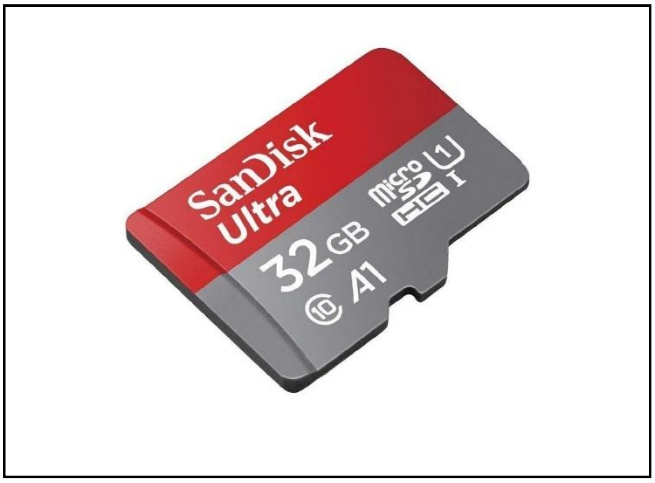

Figure 5.1.8 SD Card

CHAPTER 6

SOFTWARE DESCRIPTION

6.1. Arduino IDE:

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects. It includes a code editor which is capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

Following are the steps involved:

1. **Open Arduino IDE as shown below**

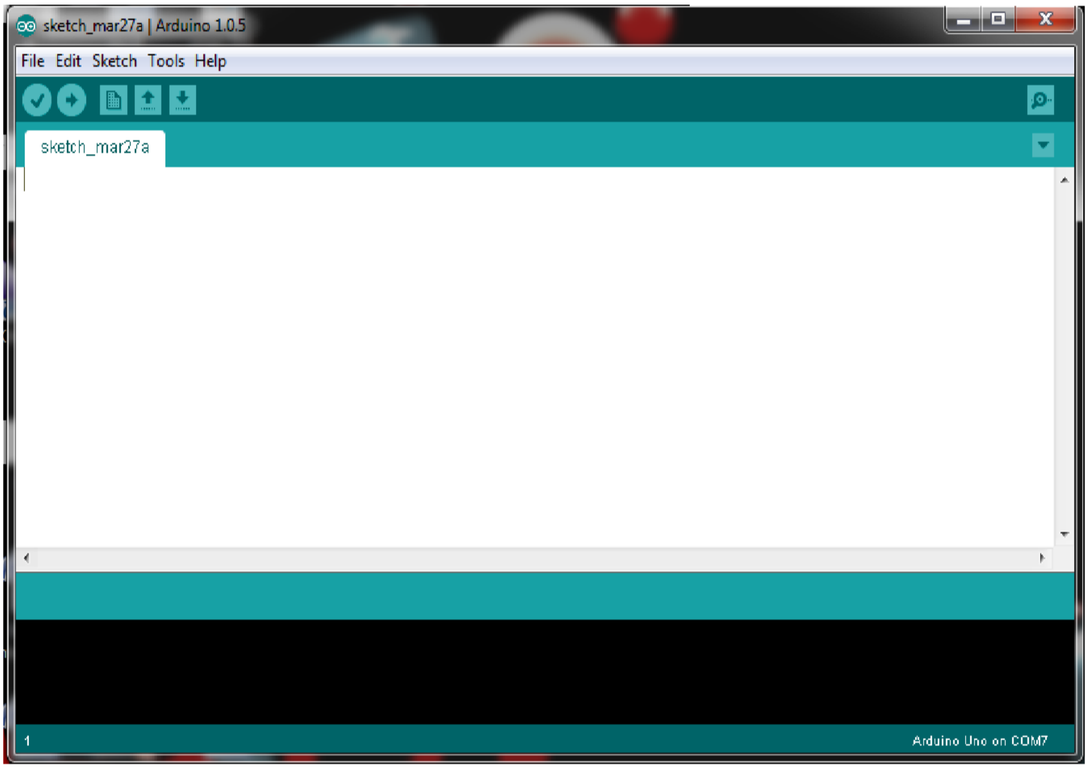

Figure 6.1.1 Arduino IDE

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much efficient. Users only need define two functions to make a runnable cyclic executive program:

setup(): a function run once at the start of a program that can initialize settings

loop(): a function called repeatedly until the board powers off

2. Select the COM Port from tools

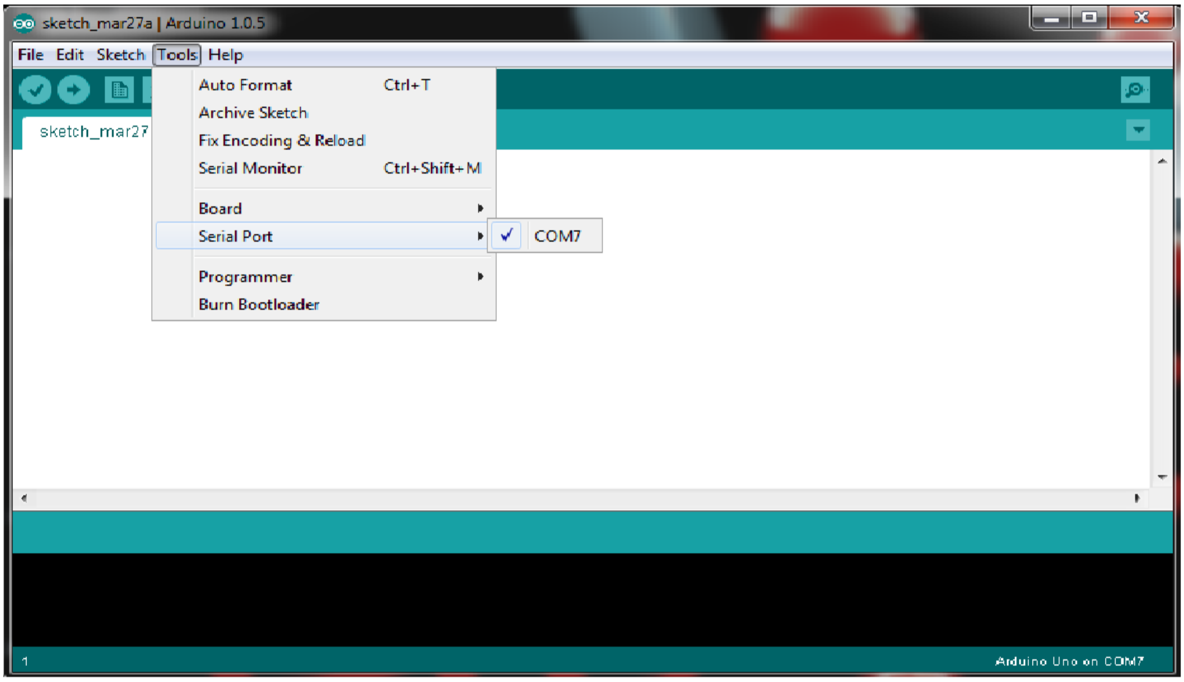

Figure 6.1.2 COM Port

3. Select the required Arduino board from Tools

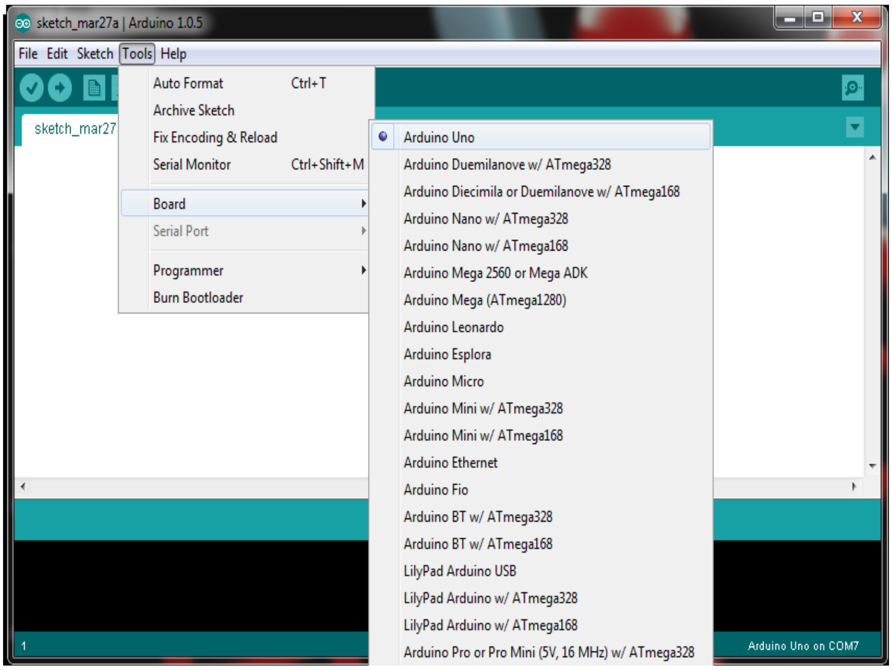

Figure 6.1.3 Tools

4. Write the sketch in Arduino IDE

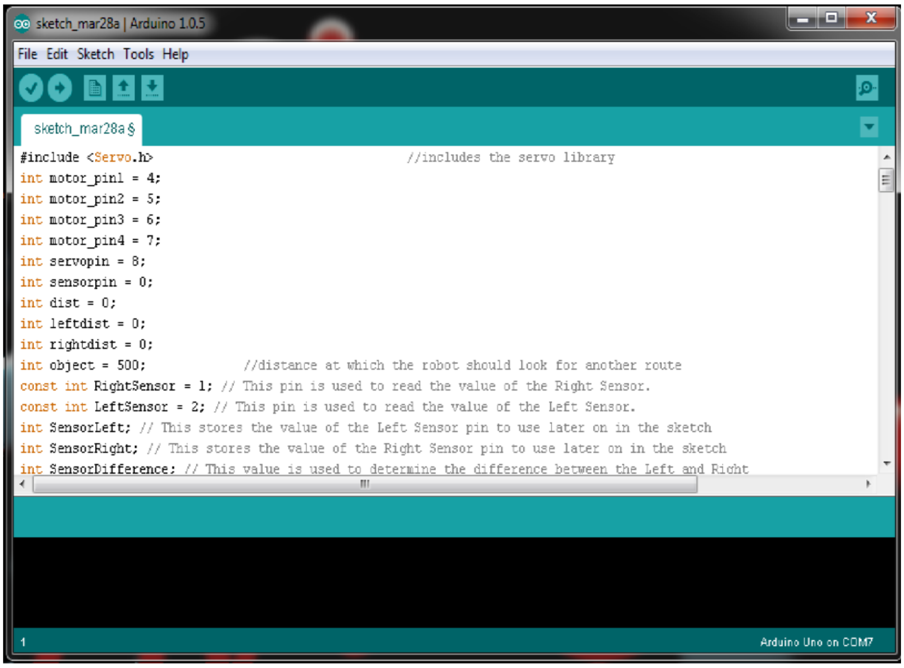

Figure 6.1.4 Sketch in IDE

5. Compile and upload the Sketch to Arduino board

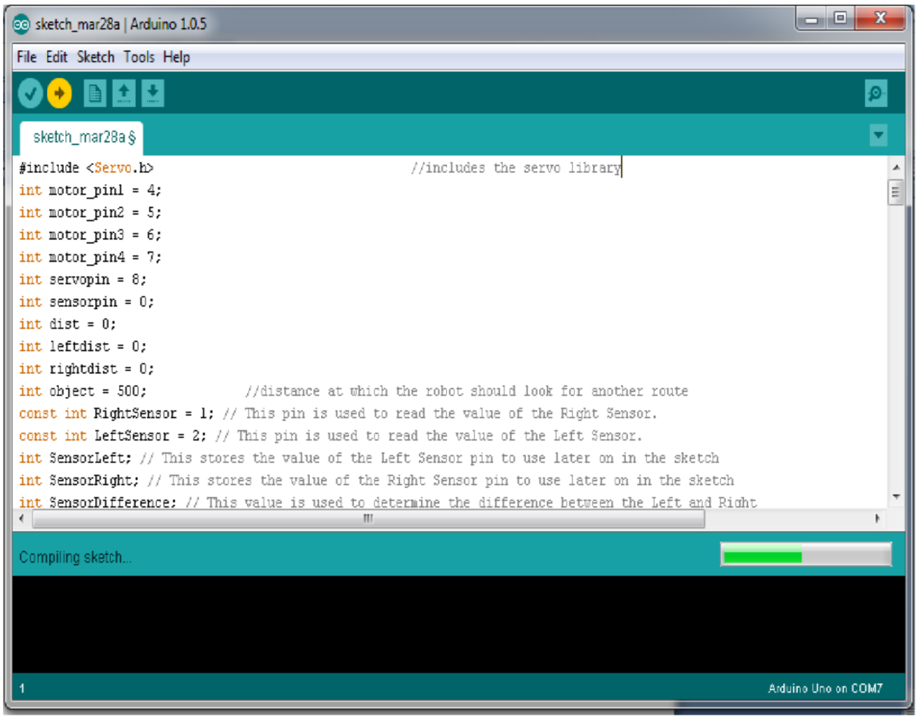

Figure 6.1.5 Compile and Upload

CHAPTER 7

RESULTS

7.1 Circuit Diagram:

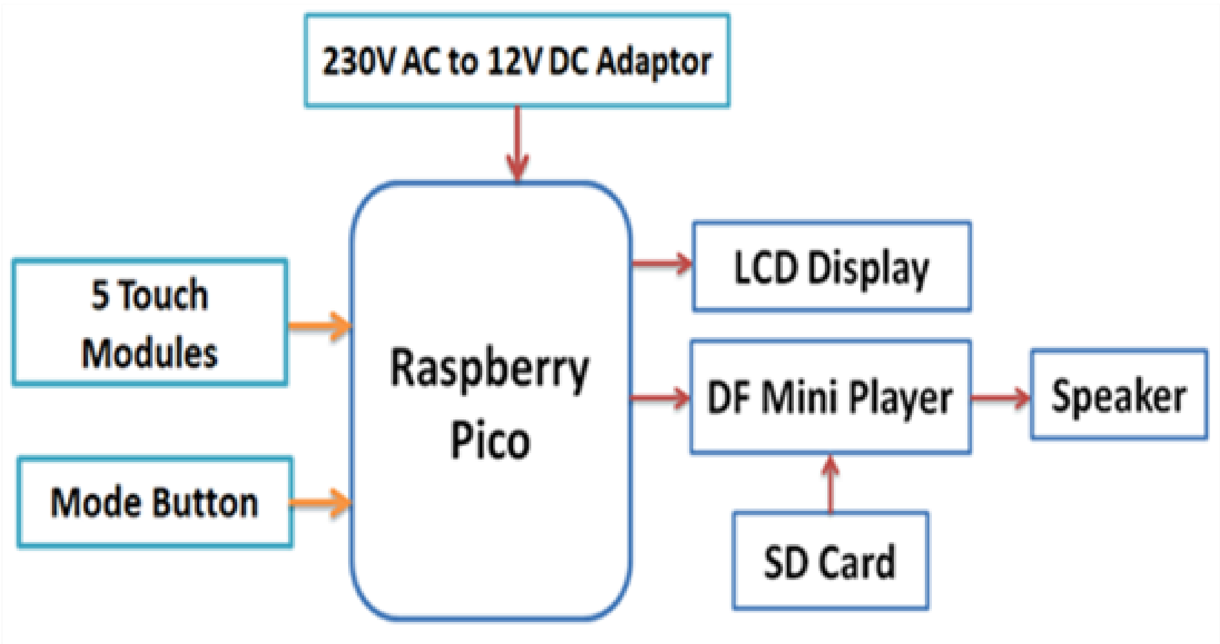

Figure 7.1 Circuit Diagram

7.2 Result:

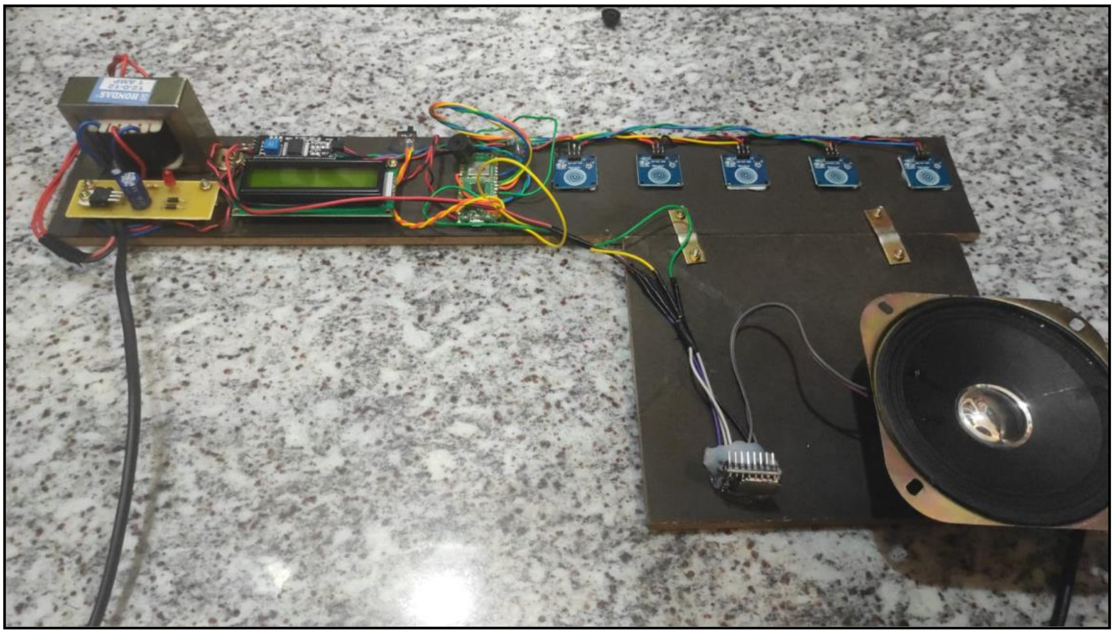

Figure 7.2 Virtual Piano Using Raspberry Pi Pico

7.3 Working

- Raspberry Pi Pico initializes system components.
- Touch sensors detect key presses.
- Signal sent to Raspberry Pi Pico.
- Raspberry Pi Pico generates audio signal.
- DF Mini Player plays corresponding note.
- LCD display updates with key pressed.
- User can switch modes via mode changer.
- Virtual piano operates continuously.
- Power off Raspberry Pi Pico to stop operation.

7.4 ADVANTAGES

- Cost-Effective
- Compact Size
- Customization
- Accessibility
- Low Power Consumption
- Community Support and Documentation

7.5 DISADVANTAGES

- Limited processing power
- Audio quality

7.6 APPLICATIONS

- Education
- Interactive Art Installations
- Entertainment and Events
- Music Learning and Practice
- Interactive Exhibits in Museums

CHAPTER 8

CONCLUSION & FUTURE SCOPE

8.1 CONCLUSION

In conclusion, the Raspberry Pico piano with touch module and DF Mini Player integration represents a compelling fusion of technology and creativity, offering users a versatile and engaging platform for musical expression. Through the seamless interaction between hardware components and software programming, this project demonstrates the potential of microcontroller-based systems to facilitate interactive musical experiences.

By harnessing the touch-sensitive capabilities of the 5-touch module and the audio playback capabilities of the DF Mini Player, users can effortlessly explore the creation of melodies, experiment with different sound modes, and immerse themselves in a world of sonic possibilities. The intuitive user interface, coupled with immediate auditory feedback, encourages experimentation and empowers users to unleash their musical creativity.

8.2 FUTURE SCOPE

The future scope of your virtual piano project using Raspberry Pi Pico and touch sensors includes the exploration of advanced features like support for multiple instrument sounds, recording and playback functionalities, and real-time effects processing, enhancing the creative possibilities for users. Additionally, integrating a graphical user interface (GUI) with interactive controls, wireless connectivity options for remote access, and compatibility with digital audio workstations (DAWs) could further elevate the user experience and extend the project's applications to collaborative music-making, educational tools, and therapeutic programs. By continually refining and expanding the virtual piano's capabilities and interfaces, your project has the potential to become a versatile and accessible platform for music creation, learning, and expression, catering to a wide range of users and use cases.

BIBLIOGRAPHY:

[1]. McPherson, A., Kim, Y., Bianchi, F., & Papetti, S. (2016). Raspberry Pi as a Platform for Musical Instrumentation. In Proceedings of the International Conference on New Interfaces for Musical Expression (pp. 348-351).

[2]. Rathod, C. S., Panchal, K. P., & Chavda, K. M. (2018). Touch Sensitive Musical Interface using Raspberry Pi. International Journal of Engineering Research and Technology, 11(4), 551-556.

[3]. Fried, L. (2017). DF Player Mini MP3 Module and Its Applications. Adafruit Industries. Retrieved from https://learn.adafruit.com/dfplayer-mini-mp3-module/overview

[4]. Gunda, A. B., Ghugare, S. S., & Baviskar, A. V. (2020). Design and Implementation of Touch Sensor-Based Piano using Arduino. International Journal for Research in Applied Science & Engineering Technology, 8(3), 1275-1280.

[5]. Fiebrink, R., Cook, P. R., & Trueman, D. (2011). Exploring Human-Computer Interaction in Digital Musical Instrument Design. Computer Music Journal, 35(1), 28-42. doi:10.1162/COMJ_a_00024

[6]. Monk, Simon. "Raspberry Pi Cookbook: Software and Hardware Problems and Solutions." O'Reilly Media, 2020.

[7]. Molloy, Derek. "Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux." Wiley, 2016.

[8]. Monk, Simon. "Make: Action: Movement, Light, and Sound with Arduino and Raspberry Pi." Maker Media, Inc., 2016.

[9]. Scherz, Paul and Monk, Simon. "Practical Electronics for Inventors." McGraw-Hill Education, 2016.

[10]. Boxall, John. "Arduino Workshop: A Hands-On Introduction with 65 Projects." No Starch Press, 2013.

APPENDIX:

Source Code (Arduino)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DFRobotDFPlayerMini.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

DFRobotDFPlayerMini player;

const int b1 = 6;
const int b2 = 7;
const int b3 = 8;
const int b4 = 9;
const int b5 = 10;

const int btn = 21;

void setup() {
  Serial.begin(115200);
  Serial1.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();
  pinMode(b1, INPUT);
  pinMode(b2, INPUT);
  pinMode(b3, INPUT);
  pinMode(b4, INPUT);
  pinMode(b5, INPUT);

  pinMode(btn, INPUT_PULLUP);
```

```
  lcd.clear();
  lcd.print("  Smart Piano  ");
  lcd.setCursor(0, 1);
  lcd.print(" Music System ");
  delay(1500);
  lcd.clear();
  lcd.print("Using Pico W");
  lcd.setCursor(0, 1);
  lcd.print(" DF Mini Player ");
  delay(1000);
  lcd.clear();
  lcd.print("Checking...");
  if (player.begin(Serial1)) {
    Serial.println("DFPlayer OK");
    lcd.setCursor(0, 1);
    lcd.print("DFPlayer OK     ");
    delay(1000);
  } else {
    Serial.println("DFPlayer Mini Failed!");
    lcd.setCursor(0, 1);
    lcd.print("DFPlayer Failed!");
    delay(1000);
  }
  lcd.clear();
  lcd.print("System Ready....");
  delay(500);
}

void loop() {
  if (digitalRead(btn) == LOW)
  {
    if (digitalRead(b1) == HIGH)
    {
      lcd.clear();
```

```
  lcd.print("Selected 1");
  delay (500);
  lcd.setCursor(0, 1);
  lcd.print("Playing Tone ");
  player.volume(30);
  player.play(1);
  delay(1000);
  return;
}

if (digitalRead(b2) == HIGH)
{
  lcd.clear();
  lcd.print("Selected 2");
  delay (500);
  lcd.setCursor(0, 1);
  lcd.clear();
  lcd.print("Playing Tone ");
  player.volume(30);
  player.play(2);
  delay(1000);
  return;
}

if (digitalRead(b3) == HIGH)
{
  lcd.clear();
  lcd.print("Selected 3");
  delay (500);
  lcd.setCursor(0, 1);
  lcd.clear();
  lcd.print("Playing Tone ");
  player.volume(30);
  player.play(3);
```

```
    delay(1000);
    return;
  }
  if (digitalRead(b4) == HIGH)
  {
    lcd.clear();
    lcd.print("Selected 4");
    delay (500);
    lcd.setCursor(0, 1);
    lcd.clear();
    lcd.print("Playing Tone ");
    player.volume(30);
    player.play(4);
    delay(1000);
    return;
  }
  if (digitalRead(b5) == HIGH)
  {
    lcd.clear();
    lcd.print("Selected 5");
    delay (500);
    lcd.setCursor(0, 1);
    lcd.clear();
    lcd.print("Playing Tone ");
    player.volume(30);
    player.play(5);
    delay(1000);
    return;
  }
}
if (digitalRead(btn) == HIGH)
{
  if (digitalRead(b1) == HIGH)
  {
```

```
  lcd.clear();
  lcd.print("Selected 6");
  delay (500);
  lcd.setCursor(0, 1);
  lcd.print("Playing Tone ");
  player.volume(30);
  player.play(6);
  delay(1000);
  return;
}
if (digitalRead(b2) == HIGH)
{
  lcd.clear();
  lcd.print("Selected 7");
  delay (500);
  lcd.setCursor(0, 1);
  lcd.clear();
  lcd.print("Playing Tone ");
  player.volume(30);
  player.play(7);
  delay(1000);
  return;
}
if (digitalRead(b3) == HIGH)
{
  lcd.clear();
  lcd.print("Selected 8");
  delay (500);
  lcd.setCursor(0, 1);
  lcd.clear();
  lcd.print("Playing Tone ");
  player.volume(30);
  player.play(8);
  delay(1000);
```

```
      return;
    }

    if (digitalRead(b4) == HIGH)
    {
      lcd.clear();
      lcd.print("Selected 9");
      delay (500);
      lcd.setCursor(0, 1);
      lcd.clear();
      lcd.print("Playing Tone ");
      player.volume(30);
      player.play(9);
      delay(1000);
      return;
    }

    if (digitalRead(b5) == HIGH)
    {
      lcd.clear();
      lcd.print("Selected 10");
      delay (500);
      lcd.setCursor(0, 1);
      lcd.clear();
      lcd.print("Playing Tone ");
      player.volume(30);
      player.play(10);
      delay(1000);
      return;
    }
  }
  delay(100);
}
```