RP2040 MACROPAD

Anna Denisova

March 2023

Contents

1 Introduction & Objective

The RP2040 Macropad is a powerful and versatile device that can be coded in different programming language to perform a wide range of functions. The device is based on the powerful RP2040 which is a micrcontroller chip designed by Raspberry Pi. It has six independent banks of RAM, a fully connected switch, and it is easily arrangeable for the cores and DMA engines to run in parallel without contention (https://www.raspberrypi.com/products/rp2040/). The Macropad itself comes with twelve keys, a rotary encoder and an OLED screen. In this project, we explore its capabilities using the Arduino IDE and C++ to create useful software.

2 Abstract

This project explores the use of the RP2040 Macropad to create a digital musical instrument. We explore how it is possible to combine and encorporate both software and harware aspects of technology making this an opportunity to learn about programming, electronics, and music all at once.

The RP2040 Macropad is implemented in a way which that the 12 keys are used to control different note pitches in a given octave. The octave itself can be increased or decreased using the rotary encoder. When a key is pressed, its note and octave value is printed to the screen of the connected laptop/computer. For convenience, there are also buttons to print newlines and sound the whole scale of the current octave at once.

This project has many real world applications. It can be used as an educational tool that helps users learn the fundamentals of music theory. Also, this can be used as a device to help with tunning where users can compare real life sounds to the sounds outputted by a computer to determine accuracy.

3 Purpose

The purpose of this project is to create software for the RP2040 Macropad to play notes of different octaves using the built in speaker.

4 Hypothesis

If I write specific code for the RP2040 Macropad then I can output different sound frequencies because of the software created.

5 Variables

Control - The RP2040 Macropad itself, additional buttons with fixed functionalities (newline, curent octave display)

Dependent - Resulting musical tone produced when a key is pressed. (Dependent on note pitch assigned to the key and the current octave set by the rotary encoder). The note names printed to the computer screen (they depend on which key is pressed)
Independent - The note pitches assigned to each key on the Macropad, The rotary encoder as it can be turned to change the octave value

6 Materials

- Full kit with all parts found here: https://www.adafruit.com/product/5128 or alternatively:
-
  - 3x4 Keys
  - Encoder
  - OLED display
  - 12 Mechanical Key Switches
  - 12 Keycaps
  - Adafruit MacroPad bottom plate
  - 3x4 Mechanical keyboard plate
  - D-Shaft Skirted Rubber Knob
  - 4 M3 5mm Machine screw
  - 4 Rubber Feet
- MacOS computer/laptop
- USB cord

7 Procedure

1. Set up your development environment (Arduino IDE) with the necessary tools to program your RP2040 Macropad.

   (a) Download the latest version of the Arduino IDE.

   (b) Add the Philhower Board Manager URL to download additional boards including the RP2040 Macropad.

   (c) In the Arduino IDE, go to Tools > Board > Boards Manager to select the RP2040 Macropad.

   (d) In the Sketch Tab, go to Port and select the corresponding port.

   (e) Now, the IDE is all set for code to be written in and run directly on the board.

   (f) Connect the board to your computer via USB.

(g) The board itself can be reset by pressing the reset button on the edge.

(h) To enter the bootloader, hold don on the rotary encoder and while continuiting to hold it, press and release the reset button. Continue holding the rotary encoder until the RPI-RP2 drive appears on your computer.

(i) Now, you are all set to write code for the RP2040 Macropad.

2. Define the notes and their corresponding frequencies using a lookup table. Some of these lists can be found here:

   - `https://pages.mtu.edu/~suits/notefreqs.html`
   - `https://mixbutton.com/mixing-articles/music-note-to-frequency-chart/`
   - `https://en.wikipedia.org/wiki/Piano_key_frequencies`
   - `https://www.liutaiomottola.com/formulae/freqtab.htm`

3. Define key-to-note mapping using an array, if statements, or a switch statements.

4. Now, code the rotary encoder to deal with octave values. With each step to the right (clockwise) increase the octal value.

5. When a key is pressed, call the tone() function to play the tone through the speaker.

6. Finally, allow printing to the computer so that the note values are outputted on the computer screen when active in any text editor (notepad, word, docs, etc.)

7. Set up any additional buttons with any additional functions.

8. Compile and upload the code to the RP2040 Macropad.

8 Experiment - The build

With a prebought RP2040 Macropad kit, the assembly was fairly quick and straight forward.

The two most popular and convenient options for which language to use are Python and C++. In this building process, C++ was used.

The board itself was coded over a span of three days with base code to go off of. First, the pitches were assigned to each key with the rotary encoder set to change the octaves. Then, using the ”Keyboard.h” library, the functionality of printing the note names to the computer was added. The additional buttons were then coded with more functionalities like adding a newline and playing the notes of a specific octave at once.

At times, the board had to be unplugged to cool off because of exessive usage as it would heat up.

Testing is a very crucial part. As new functions are added, testing is necessary to confirm that whatever was added actually works as intented.

9 Results & Analysis

The creation process did not always run smoothest. Of course, there were bugs or some of my plants did not go as planned which forced me to code the functionalities alternatively. For example, I had to figure out how to map out the frequencies of waves to their corresponding note values and their octaves.

Generally, there were many different ways for going about how to design the code. There isn't always one specific way to do something but rather an array of different options and paths to take. Of course, some paths may be more efficient and generally better in principle.

One of the biggest challenges was the fine-tuning each of the different functionalities coded. The tone generator had to produce the pitches at the correct frequencies for each octave, the position of the rotary encoder had to be carefully monitored and checked, and the structure of the code had to be logical and efficient. However, through testing and research, I was able to overcome these challenges.

””” From a technical standpoint, the RP2040 Macropad with 12 keys and rotary encoder is a well-designed and implemented project that demonstrates a good understanding of hardware interfacing, software development, and embedded systems. The use of PWM and tone generator libraries to produce different pitches and frequencies for each note and octave shows a deep understanding of the underlying principles of sound generation and modulation.

Furthermore, the addition of extra button functionalities such as printing a newline and playing the whole scale showcases a good understanding of software development and user experience design. The ability to print debug information to the computer screen using USB serial communication is also a valuable feature that improves the usability and functionality of the Macropad.

Overall, the RP2040 Macropad with 12 keys and rotary encoder is a well-implemented project that demonstrates a good understanding of both hardware and software development. The ability to produce different pitches and frequencies for each note and octave, along with the additional button functionalities, makes it a versatile and useful tool for musicians and developers alike. ”””

10 Conclusion & Application

Using the built in speaker, code was made to map the keys of the RP2040 Macropad to different musical notes and to make the rotary encoder into a way to increase/decrease the current octave.

The RP2040 microcontroller chip designed by Raspberry Pi is high performance, low cost, and has flexible I/O being very useful in real life applications as it can speak to almost any external device (https://www.raspberrypi.com/products/rp2040/).

In this project, the microcontroller is directly connected to an external laptop to transfer signals through a USB cord.

There are several practical applications for this project. One such application is an educational tool to help users learn the basics of music theory. Also, the RP2040 Macropad can be used as a tuning device, allowing users to compare real-life sounds to the computer-generated sounds in order to determine accuracy.

///////

Completing this project is a nice way to gain educational experience and a deeper understanding of microcontrollers.

I gained a deeper understanding of the fundamentals of microcontrollers. I learned how to program code for the buttons, rotary encoder, and tone generator. I also learned how to use many different libraries such as the Adafruit NeoPixel library, the Rotary Encoder library and the Keyboard library.

The ability to assign different note pitches to each key and allow for octave control through the rotary encoder makes the RP2040 Macropad a useful musical tool. As well, the ability to print the note names to the computer screen and play the corresponding tones in real-time creates a compelling experience to users.

There are many interactive functions that allow users to experiment and use this advice in practical ways.

It is important to mention that the abilities of the RP2040 Macropad do not end here. There are many more different projects to create and experiment with!

11 References

https://www.adafruit.com/product/5128

12