

RP2040 MACROPAD

Anna Denisova

March 2023

Contents

1	Abstract	2
2	Introduction & Objective	2
3	Purpose	2
4	Hypothesis	2
5	Variables	2
6	Materials	3
7	Procedure	3
8	Experiment - The build	5
9	Results & Analysis	6
10	Conclusion & Application	7
11	Libraries Used	8
12	Bibliography	8

1 Abstract

This project explores the use of the RP2040 Macropad to create a digital musical instrument. We test the possibility of combining and incorporating both software and hardware 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 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 was chosen because it has many real world applications. It can be used as an educational tool that helps users learn the fundamentals of music theory. Also, it can be used as a device to help with tuning where users can compare real life sounds to the sounds outputted by a computer to determine accuracy. Overall, the RP2040 was successfully coded with software that works as intended.

2 Introduction & Objective

The RP2040 Macropad is a powerful and versatile device that can be coded in different programming languages to perform a wide range of functions. It is based on the high performance RP2040 microcontroller chip designed by Raspberry Pi. It has a large chip memory, a symmetric dual-core processor complex, deterministic bus fabric, and a rich peripheral set [1]. 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.

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
- Additional buttons with fixed functionalities
- Computer, USB cord

Dependent

- 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 (depend on key pressed)

Independent

- 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 cable

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 of the board.
 - (e) Now, the IDE is all set for code to be written in and run.
 - (f) Connect the board to your computer via USB.
 - (g) The board itself can be reset by pressing the reset button on the edge of it.
 - (h) To enter the bootloader, press the rotary encoder and while continuing holding it down, press and release the reset button. Continue holding the rotary encoder down until the RPI-RP2 drive appears on your computer.
2. No, you are all set to write C++ code.
 3. Define the notes and their corresponding frequencies using a lookup table. Some of these tables 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>
 4. Define key-to-note mapping using an array, if statements, or a switch statement.
 5. Code the rotary encoder to deal with octave values. With each step to the right (clockwise) on the rotary encoder, increase the octal value.
 6. When a key is pressed, call the tone() function to play the note through the speaker.
 7. 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.)
 8. Set up any additional buttons with any additional functions.
 9. Compile and upload the code to the RP2040 Macropad.

8 Experiment - The build

With a pre-bought RP2040 Macropad kit, the assembly was fairly quick and straightforward.

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

Some safety precautions include properly handling electrical components so as not to damage them, having the correct connections throughout the device before applying power to prevent shorts or other issues, being mindful of heat being generated by the components, using proper documentation of the circuit and code, and following other various best practices.

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 excessive usage causing it to heat up.

Testing was a very crucial part as well. As new functions were added, testing was necessary to confirm the correctness of newly added code.

Here is the final program:

9 Results & Analysis

Generally, there were many different ways for going about how to design the program. There isn't always one specific way to achieve a function/task 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 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. Overall, from a technical standpoint, the RP2040 Macropad is a well-designed and implemented project that demonstrates good understandings of hardware interfacing, software development, and embedded systems. The use of tone generation to produce different pitches and frequencies for each note and octave teaches the creator the underlying principles of sound generation and modulation.

10 Conclusion & Application

Using the built in speaker, a program was created to map the keys of the RP2040 Macropad to different musical notes with the rotary encoder making the octaves adjustable.

The RP2040 microcontroller chip designed by Raspberry Pi has high performance, a low cost, and flexible I/O. It can communicate with almost any external device making it very useful in real life applications [2]. In this project, the microcontroller is directly connected to an external laptop to transfer signals through a USB cable.

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. The ability to print the note names to the computer screen and play the corresponding tones in real-time creates a compelling user experience.

Completing this project is a reliable way to gain educational experience and a deeper understanding of microcontrollers. Personally, I gained more experience with understanding the fundamentals of the RP2040. I learned how to program code for the buttons, rotary encoder, and tone generator. Also, I learned how to use many different code libraries such as the Adafruit NeoPixel library, the Rotary Encoder library and the Keyboard Library.

However, there are also many potential sources of error that may occur when developing such a project. There could be issues with signals which could lead to notes being off pitch or having inconsistent timing. The rotary and button errors may fail to register input correctly and there could also be communication issues when connecting the Macropad to the laptop/computer. Finally, the

most common error that could occur is within the code. These include bugs, logic, or any syntax errors that could cause unexpected behavior or crashes.

Of course, the abilities of the RP2040 Macropad do not end here. There are still many more different projects to create and experiment with!

11 Libraries Used

The code uses the following libraries:

- `Adafruit_SH110X.h`
Copyright (c) 2012, Adafruit Industries. All rights reserved.
License: BSD
- `Adafruit_NeoPixel.h`
Copyright (c) 2007, Free Software Foundation
License: LGPL-3.0
- `RotaryEncoder.h`
Copyright (c) 2005-2014 by Matthias Hertel, <http://www.mathertel.de/>
License: BSD
- `Wire.h`
Copyright (c) 2006 Nicholas Zambetti. All right reserved.
License: LGPL-3.0
- `Keyboard.h`
Copyright (c) 2015, Arduino LLC
License: LGPL-3.0

12 Bibliography

1. <https://datasheets.raspberrypi.com/rp2040/rp2040-product-brief.pdf>
2. <https://www.raspberrypi.com/products/rp2040/>
3. <https://www.adafruit.com/product/5128>
4. <https://learn.adafruit.com/adafruit-macropad-rp2040>

5. <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-macropad-rp2040.pdf>
6. <https://www.arduino.cc/>
7. <https://pages.mtu.edu/~suits/notefreqs.html>
8. <https://mixbutton.com/mixing-articles/music-note-to-frequency-chart/>
9. https://en.wikipedia.org/wiki/Piano_key_frequencies
10. <https://www.liutaiomottola.com/formulae/freqtab.htm>