Integrating internet resources and inexpensive materials to evolve a Raspberry Pi into a digital button sound machine.

Benjamin Guitreau
Center for Communication and Technology
Baton Rouge, Louisiana
bguitr1@tigers.lsu.edu

ABSTRACT

The goal of this paper is to provide the details involved in designing and building a digital musical enclosure that uses a button matrix as the keys. The idea is to allow for an instrument that can be replicated by hobbyists, students, and professionals alike by using inexpensive parts, materials, and resources that are only a few keystrokes away. The project is scalable in any direction and is not limited to producing audio signals.

Keywords

NIME, proceedings, Raspberry Pi, Sparkfun

1. INTRODUCTION

The materials in the construction are inexpensive and accessible from online retailers such as Sparkfun[1], while the design is straight-forward and easily reproducable or modified.. The basic programming to get the instrument working is minimal which allows for anyone to be able to write the code, while allowing for more advanced users to create complex scores.

2. DESIGN

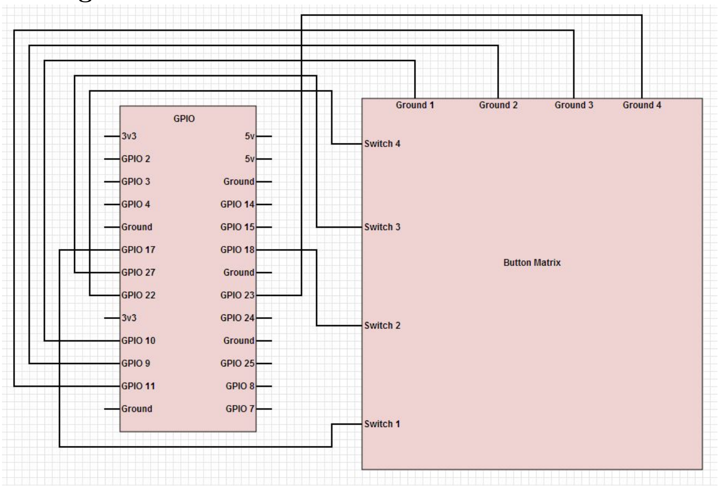
2.1 Hardware

The design of the box is based upon an enclosure built by Berdahl[2]. Using BoxMaker[3], the builder is able to initiate the design process by inputting the dimensions and material thickness of the box. Using the downloaded design, the builder is able to edit the design with either Inkscape, CorelDraw, or Adobe Illustrator which allows for the creation of a project that can then be uploaded to Ponoko[10]. Depending upon the material of choice, Ponoko will cut the pieces for the box and ship them to the builder.

The electronic hardware was all be purchased from the Sparkfun website for less than $70. A Raspberry Pi[11] is mounted inside of the box keeping the GPIO pins accessible. A button pad breakout board[6] is mounted on the top of the box, with a silicon rubber button pad[7] placed on top of the breakout board. Soldered to the board are 16 diodes[8] and 8 breakaway header pins[5]. Female to female jumper wires[9] are used to connect the header pins to the GPIO pins that will be used in the construction of the instrument.

The GPIO pins and the breakout board are connected per the following figure.

Figure 1: GPIO-Breakout board schematic



2.2 Software

A python script, that is started during the bootup sequence of the Pi, is controlling the reading of the button matrix. The builder must install either python3 or python2 (python3 was used for the project and all source code available will be using python3) and RPi.GPIO[4] to be able to use the instrument. The Pi has internal pull-up/pull-down resistors, which are necessary to discern the button being pressed. In the snippet of code below, 3 lists are created. The first list is the matrix allocation of buttons, while the other two lists contain the GPIO pin number connected to the respective row or column.

```python
import RPi.GPIO as GPIO

class ButtonMatrix():
  BUTTON_MATRIX = [
    [1,2,3,4],
    [5,6,7,8],
    [9,10,11,12],
    [13,14,15,16]]
  ROW = [22,27,18,17]
  COLUMN = [10,9,11,23]
```

The script initially sets GPIO column pins as output with a low value. The GPIO row pins are initially set as input with pull-up resistors. When a button is pressed, the row value is read and then the column and row settings are reversed, except instead of a low value, the row pins are set with an output at high value and the column pins are set as an input with pull-down resistors. The column value is then read and the button matrix value is calculated using the two dimensional indices of the matrix list.

Pure Data extended, or pd-extended, is running on the Pi for musical/audio generation. Python and pd are communicating using the pdsend and pdreceive. In this setting, there is no discernible latency between button presses.

3. CONCLUSIONS

The builder can can spend under $100 (US) and less than a few hours of time to create a digital musical instrument. By using Ponoko, anyone can design and build the box without worrying about having access to a laser-cutter, which allows for a plethora of opportunities for younger students and hobbyists to expand upon this project.

4. ACKNOWLEDGMENTS

The author would like to thank Edgar Berdahl for the opportunity and support in the design and implementation.

5. REFERENCES

[1] Sparkfun.

[2] E. Berdahl. Howtobuildapoweredloudspeaker from modernlutherie. https://ccrma.stanford.edu/~eberdahl/7745/HowToBuildAPoweredLoudspeaker.pdf, Feb. 2014.

[3] R. Bhargava. Boxmaker. http://boxmaker.rahulbotics.com/, 2013.

[4] B. Croston. Rpi.gpio. https://pypi.python.org/pypi/RPi.GPIO, 2014.

[5] Sparkfun. Break away headers - straight. https://www.sparkfun.com/products/116.

[6] Sparkfun. Button pad 4x4 - breakout pcb. https://www.sparkfun.com/products/8033.

[7] Sparkfun. Button pad 4x4 - led compatible. https://www.sparkfun.com/products/7835.

[8] Sparkfun. Diode small signal - 1n4148. https://www.sparkfun.com/products/8588.

[9] Sparkfun. Jumper wires premium 6" f/f pack of 10. https://www.sparkfun.com/products/8430.

[10] Sparkfun. Ponoko. http://www.ponoko.com/make-and-sell/how-it-works.

[11] Sparkfun. Raspberry pi - model b. https://www.sparkfun.com/products/11546.