

Audio Extravaganza

Martin Barker

Team 26 CS461 Fall 2018

1 My Role:

Software writer / tester / community outreach

2 What we are trying to accomplish:

The purpose of this project is to create a functional looping / sound modulation pedal for live performances that lives up to Moog's legacy. An effects pedal has many different purposes for musicians, for use in instances such as recordings and live performances. The world of hardware musical devices has many different communities with nostalgia for analog instruments and vintage devices. Our effects pedal will try to pay homage to the legacy of looping/modulation pedals, while also striving for improvement and innovation. We will also focus on usability, and creating a functional pedal which can hold up to the pressures of live performance.

3 Item 1: Button Subsystems

This section concerns the button inputs we will be using for our pedal, such as on/off buttons, dials, and various methods for our single board computer to receive input. The decision criteria I will be looking at includes which option gives the most flexibility and simplicity for getting our computer to recognize our input.

3.1 Tech option 1: Reading input directly from button via python

Example projects that involve using buttons with Raspberry Pi's vary in difficulty from beginner to expert, and there are multiple ways to connect a button to a Raspberry Pi. One such way involves using the Raspberry Pi's general purpose input output ports (GPIO), and connecting it to a momentary tactile push button switch by creating a simple circuit [1]. This method requires you

to make small wire connections using the Pi's IO board, which on its own is a delicate connection.

A button connected to the Pi's IO board works by sending a low (0V) signal when not pushed, and sending a high (3.3V) signal when the button is pushed. The button's state can be read by a Python program which reads the GPIO pin status. The Raspberry Pi GPIO Python Module required to accomplish this can be downloaded by running the command:

```
$ sudo apt-get install python-rpi.gpio python3-rpi.gpio
```

The program you write will read the button's input so long as the program is running. In the program you specify which pin is the input/output, and can then read the button's state and make corresponding changes. Reading in the state of an input device using a python program seems like one potential way for us to interact with input / output on our effects pedal, seeing as there are many different python libraries and packages to communicate with audio software.

3.2 Tech option 2: Reading input using RC circuit

By hooking up an RC circuit to our board we would have more GPIO pins at our disposal. Each pin on the Raspberry Pi board has a software configurable pull-up and pull-down resistors. You are able to configure the resistors so that one or none or neither are enabled using the pull_up_down parameter. If this parameter is not included, the input will be left in a state of 'floating', where the value cannot be relied upon and it may drift between high and low depending on electrical noise. Knowing how to avoid 'floating' states in our pedal will be very important, as 'floating' values can cause unreliable behavior. It will be important to use the GPIO setup and pull_up_down values in order to mitigate floating values and ensure our inputs / outputs are predictable. [2]

3.3 Tech option 3: Mapping input from mouse / keyboard

One interesting feature I found for SuperCollider on the composerprogrammer website is the ability to manipulate sound properties in SuperCollider using the keyboard as input[3]. For example, the following command uses keyboard input to change sound:

```
//key code 0 = 'a' key  
{ SinOsc.ar(800, 0, KeyState.kr(0, 0, 0.1)) }.play; //Server-side
```

This feature could make getting input from hardware buttons / knobs easier, if we treated the input as a keyboard button press and interpreted it that way into SuperCollider. Mouse input can also be used to change sound properties, which means that the values of hardware knobs / dials / buttons could also potentially be mapped to a mouse input for use within SuperCollider. Another useful resource I found was on the arduino forums[4], where somebody is asking for help on a SuperCollider path making noise when an arduino button is pressed.

In the forum posts, SuperCollider code is posted which features the following line to connect to an arduino device:

```
p = ArduinoSMS("/dev/tty.usbserial-A4001nK8", 115200); ///connecting to Arduino
```

This will be a useful resource for reading input from our hardware into our music software, as there are resources online with code such as the forum post above that give examples of how the arduino / SuperCollider code should communicate.

3.4 Conclusion

While mapping keyboard / mouse inputs would be the simplest, using an external RC circuit would give us more potential inputs to use. Using this approach would provide its own set of challenges, such as mitigating electrical interference, but ultimately it is a method of input which has extensive documentation online, making it a solid choice for our single board computer.

4 Item 2: Metrics for testing

This section will cover the metrics we will use to test our device. The decision criteria for this section will depend on how understandable the metric is, how difficult the metric is to gather, and how much useful information we are able to gather from the metric.

4.1 Testing based on user technical experience

To test our pedal based on a user's technical experience we would have to choose our subjects based on how much previous experience they have using guitar effects pedals. In this case the metrics we would be looking at when choosing a subject is how long they have used effects pedals, what they would rate their proficiency with effects pedals on a scale of 1 to 10, and what types of effects pedals have they used / feel most comfortable with. These metrics would give us insight onto their feedback about our pedal, such as how easy our looping feature is to somebody with extensive experience using looping pedals.

4.2 Testing based on user musical experience

Determining test subjects based on a user's musical experience would tell us how intuitive our pedal is to use. Our subjects in this case would be categorized based on how much musical experience they have in general. The subjects would range in terms of technical experience, but the main focus would be on user experience. Metrics such as x , x , and x would tell us how intuitive and learnable our effects pedal interface is.

4.3 Testing based on performance durability/reliability

Reliability is an important aspect we need to test for our device, and in order to accurately test it we will need to measure the metrics of how long it takes for our device to boot and shut down, as well as complete basic tasks. As soon as we start implementing our effects in hardware, we should take regular measurements in seconds of how long these various aspects take to complete. These measurements can be stored in an online spreadsheet, and updated regularly. This would give us good metrics on how our device boot / shutdown times either improved or worsened over the length of our development. To simulate stressful conditions in a testing environment we can leave our effects pedal running for multiple hours, as well as repeatedly turn on and off an effect. These tests would tell us how well are pedal would hold up in performance situations, but would take multiple hours to accurately replicate.

4.4 Conclusion

Testing based on user technical experience would give us the most valuable information, as there is valuable feedback to be gained from enlisting musical users with varying technical experience. Corvallis has a very active music scene and we have two musicians within our project group. So enlisting musicians to test our pedal and interpreting their responses while understanding their technical background would give plenty of feedback for us to evaluate.

5 Item 3: Methods of getting power into an Raspberry Pi

This section will look at what some of the options are for powering our Raspberry Pi effects pedal. The decision criteria will be how much power the tech option provides, how durable the connection it is, and how time/resource intensive the method is to use. The Pi 3 has a recommended power supply of 5.1V at 2.5A. [5]

5.1 Battery power

Portable smartphone batteries can be used to charge the Pi, and would require the purchase of some external battery which meets the Pi's power requirements. This purchase would likely be within the range of one hundred dollars, but would give our effects pedal more portability and rechargeability. Using more common batteries such as AA or AAA would require the addition of a battery port connected to our device, which would be another thing to focus attention on getting to work, and would have to be worked on the ensure durability.

5.2 Wall power

The official Raspberry Pi power supply unit offers 5.1V at 2.5A, which is enough to power the board under most circumstances. Using an official PSU also gives more confidence in avoiding technical malfunctions with unregulated power supplies. The official power supply is available for only nine dollars, which makes it a very affordable option. Using the official Pi power cable would also provide a sturdy connection.

5.3 USB power

It is possible for us to draw power to the Raspberry Pi from a USB cable, either connected to a computer or a USB to outlet converter. While supporting this approach is more flexible and convenient for people, there are issues with how much power a standard usb cable draws. The Pi 3 has a recommended power supply of 5.1V at 2.5A, while USB 3.0 is only capable of delivering up to 0.9A [6].

5.4 Conclusion

Using the official Raspberry Pi power cable would give us plenty of power for our board and any peripherals we use, as well as provide a reliable and durable connection. While this approach does sacrifice some portability, most effects pedals available on the market have wall outlet connections, so running our pedal off wall power wouldn't be too surprising to a performer who already uses effects pedals.

References

- [1] "Using a Push Button with Raspberry Pi GPIO." Raspberry Pi HQ, 9 Feb. 2018, rasberrypi.org/forums/viewtopic.php?p=14092246.
- [2] "4 Ways To Get Raspberry Pi To Read Multiple Analog Input Sensors!" Gadget Explained, www.gadgetexplained.com/2016/05/4-ways-to-get-raspberry-pi-to-read.html.
- [3] Nick Collins, composerprogrammer.com/teaching/supercollider/sctutorial/4.1%20Interaction%201.html.
- [4] Installation Using Arduino and SuperCollider, forum.arduino.cc/index.php?topic=38344.0.
- [5] Saville, Richard. "10 Ways to Power Your Raspberry Pi." Lifewire, Lifewire, www.lifewire.com/ways-to-power-your-raspberry-pi-4092246.
- [6] Lendino, Jamie. "How USB Charging Works, or How to Avoid Blowing Up Your Smartphone." ExtremeTech, 21 Apr. 2017, www.extremetech.com/computing/115251-how-usb-charging-works-or-how-to-avoid-blowing-up-your-smartphone.