What do I want to build?

I want to build a device that can tell me how sharp or flat the pitch of my instrument is. In other words, I want to be able to make a tuner to attach onto my instrument. It should be handheld, portable, be able to clip onto the bell of my instrument and communicate to me how far off my pitch is.

What do I hope to achieve through this project?

* Learn more about audio engineering and signal processing with audio waves
* Learn more about the mechanics of sound waves, specifically how notes can be “out-of-tune"
* Write an algorithm that can analyze the certain pitch of a sound wave and determine how far off it is from the normal pitch

What is the timeline for this project?

Span of 1-2 months. I hope to research throughout the entirety of this project. I want to focus heavily on research initially, and later move onto system architecture and actual development in the latter half.

Resources

The Hive as my makerspace

Research Log

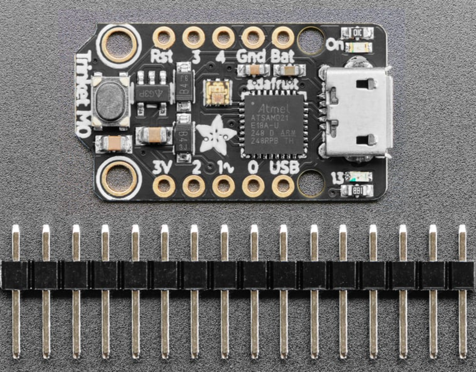
11/13

Initial Research: What the hell am I making?

My initial research consisted mainly of figuring out what this tuner is going to be. Is it going to be battery powered or connected to my machine? How am I going to take in audio input? Am I going to use a raw microphone or one that is fixed to a breakout board? How will I display the results to the user?

I spent most of my time searching through Amazon and Adafruit seeing what microcontrollers were available since this is the basis of what the entire product will rest on. I believe the best choice for me is the Adafruit Trinket M0. Here are a few reasons as to why I chose the Adafruit Trinket M0

* Size
  + The trinket M0 is incredibly small. It uses a micro usb as its flasher. It is pretty much the size of a quarter
* Capabilities
  + The Trinket M0 uses the AT SAMD21 microchip.
  + The breakout board is capable of I2C communication & it supports 5 GPI/O pins
  + The microphone module I have has an analog output; this output can be connected to any GPIO pin as there is an internal ADC in the Trinket M0
* Power Supply
  + The power supply for this MCU is very small. It takes in 3.3V - 6V and steps it down by its own voltage converter to a steady 3V
  + There is a 3.3V output pin that can be used to power other peripheral components
  + I plan on using 3 triple A batteries wired in series (1.5\*3 = 4.5) to power the MCU and other components as necessary



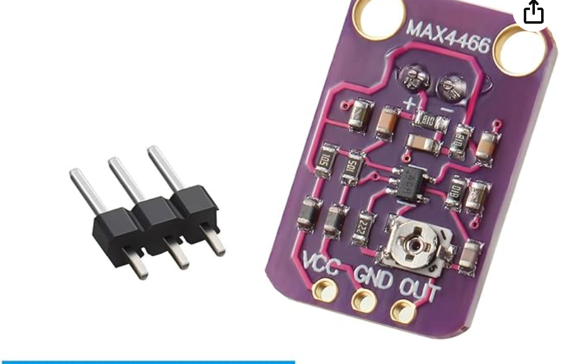
11/16

After figuring out what MCU I am going with, I needed to figure out my most other crucial component. How am I going to capture audio signals? Obviously, I need a microphone; but do I go with a bare bones microphone? Or should I go with one prefixed onto a development board? My next area of research focused on the pros and cons of each choice. I soon found out that going with a raw microphone unit was simply not a good choice. Like how I chose my microcontroller, I opted for a microphone module prefixed onto a development board simply because I did not want to deal with all the intricacies that go into wiring all the passive components to the microphone to get a clear analog signal.

It becomes apparent the advantages of choosing a development board rather than a bare-bones microphone because of how many passive components are needed. Confer the image below which shows the back of the microphone module I am about to buy.



As you can see, there is an array of resistors, capacitors, and Schottky diodes all mounted to the board in an intricate schematic. Deriving this from scratch based on my own research would easily add ~2 weeks to the project timeline. Although this would be beneficial research (as all research is beneficial in some respects), this is not the scope of my project. My project is aimed to teach me how to determine variance of a particular pitch from a standard one (in essence, a tuner). Nonetheless, I decided to go with the following microphone module: Microphone Amplifier Module MAX4466



Okay. Now that I have the two most integral components, I’m still missing the thing that will make this all work. Power. How will I power this system?

Contrary to popular belief, power is not simply determinant on the supply voltage needed for all the components. Based on my experience with RoboJackets, the power budgeting of an electrical system is intricate and involves a lot of moving numbers including the nominal supply voltage, nominal current rating, peak voltage, and peak current rating.

We break the power budgeting for each component. Consider first the microcontroller we are using. The nominal supply voltage for the Adafruit Trinket M0 is 3.3V to 6V where it has an internal voltage regulator to step down the voltage to 3V (https://cdn-learn.adafruit.com/downloads/pdf/adafruit-trinket-m0-circuitpython-arduino.pdf). Thorough research has left me unable to determine the nominal current for the microcontroller itself. We can assume that since the board uses the SAMD21 microchip from Atmel, it belongs to a family where "more than 10mA will probably kill them" (<https://forums.adafruit.com/viewtopic.php?f=8&p=618630#:~:text=The%20SAMD21%20microcontroller%20uses%203.3,we%20suggest%20staying%20below%207mA>.) As a result, we will assume the peak current draw for the MCU itself is 7mA. The 3.3V supply pin on the board is rated to pass up to 500mA of current through it. This number refers to the peak current draw of the board. As a result, the power consumption at peak current draw is going to be as follows:

Although this calculation is robust, it is a complete overestimation of what can happen for the whole microcontroller. Because we are connecting the microphone module to the 3.3V power supply, the peak current draw for the 3.3V power supply can only max out at the peak current draw of the microphone module which is 24 microamps. The revised equation is as follows:

Because the power consumption for this project is so incredibly low, we are fine with powering the system off of 3 AAA batteries connected in series for a total supply voltage of 4.5V.

**Circuit Design**

A screenshot of a computer

Description automatically generated

**V1 Circuit Schematic**

Circuit analysis and research:

11/17/2023

Consider my initial schematic for the circuit. Let’s break it down part by part. The microcontroller is, as we all know, the heart of the circuit as it is the “brain” of the system. To power the microcontroller, we need an input voltage to the “BAT” line of 3.3V – 6V. As stated above, I will be using 3 AAA batteries to drive this circuit. This is as to why I have three voltage sources. Specifically, to produce a total voltage of 4.5V, I wired each 1.5V battery in series. The positive terminal of this unit is fed into the “BAT” line of the Trinket M0. Consequently, I connected the negative terminal of the power unit to the “GND” pin of the MCU.

As for the microphone module, we derive its power from the 3.3V pin of the Trinket itself which can provide a steady, clean voltage (via built-in decoupling capacitors) and up to 500mA of current; these specs are overkill as the microphone module will draw 24uA nominally. The “OUT” pin of the microphone module is an analog output that provides the data we need. We connect this pin to any GPIO pin of the Trinket (in this case, we use pin 0). However, like I said, this signal is an analog signal, meaning it is a continuous voltage. To perform analysis over this signal, we must interpret it as discrete values. This is where the analog to digital converter (ADC) is required. One of the benefits of the Trinket is that all GPIO pins are capable of analog input, meaning they have a built-in ADC (12-bit analog input). For now, we will only deal with how to physically connect the components together. This topic of analyzing the audio input dives far deeper, which will be covered in the codebase architecture. The process of converting the signal from analog to digital logic is one of two steps in interpreting the pitch of an audio.

What is an analog to digital converter even?

11/20/2023

The analog to digital converter, more commonly known as the ADC, is an integrated circuit that takes an analog signal, such as a voltage or a current, and assigns it a “digital value.” But what does digital value even mean? It refers to the lowest level in which information is encoded in discrete values: binary. The ADC takes a floating voltage or current at different magnitudes and converts it into discrete binary values.

What came as a surprise to me was the fact that ADCs are fully integrated circuits, and not software algorithms. Essentially, the analog signal which comes in at a continuous-time and continuous amplitude wave function is converted into a set of discrete values which describe the amplitude of the voltage, a process known as quantization.