Robotic Xylophone with Adafruit Grand Central

Created by Dano Wall

https://learn.adafruit.com/robotic-xylophone-with-circuit-python

Last updated on 2023-08-29 03:58:41 PM EDT
Table of Contents

Overview 3
- Your New Musical Butler
- The Electronics
- Materials
- Play On

Solenoid Circuit 8
- Gather what you'll need
- The N-channel Power MOSFET
- Soldering the Circuit
- The Plug
- Connect JST Extension Cables
- Add Headers

Assemble the Frame 22
- Height Adjustable

CircuitPython 25
- Getting Familiar
- Demo Code
- Troubleshooting
Overview

Your New Musical Butler

This robotic musician uses miniature solenoids to tap out melodies on a toy xylophone. The Adafruit Grand Central M4 controls each of the eight solenoids using CircuitPython, one of the fastest growing programming languages, specifically designed to simplify experimenting and learning.

This guide demonstrates how to build your own robotic xylophone rig, and how create your own melodies in CircuitPython.

The Electronics

Adafruit Grand Central M4 Express featuring the SAMD51
Are you ready? Really ready? Cause here comes the Adafruit Grand Central featuring the Microchip ATSAMD51. This dev board is so big, it's not...
https://www.adafruit.com/product/4064
**Mega protoshield for Arduino**

This is an extra large proto-shield, specifically designed for the Mega! It comes with PCB, two buttons and a full set of stacking headers. You can stack this shield easily since it...

https://www.adafruit.com/product/192

**Mini Push-Pull Solenoid - 5V**

Solenoids are basically electromagnets: they are made of a coil of copper wire with an armature (a slug of metal) in the middle. When the coil is energized, the slug is pulled into the...

https://www.adafruit.com/product/2776

**N-channel power MOSFET**

When you need to switch a lot of power, N-channel MOSFETs are best for the job. These FETs can switch over 60A and 30V and are TO-220 packages so they fit nicely into any breadboard or...

https://www.adafruit.com/product/355

**1N4001 Diode - 10 pack**

This here is a 10 pack of the classic 1N4001 power blocking diode. These are good for reverse polarity protection (put it between your DC power jack and circuitry to avoid a...

5V 2A (2000mA) switching power supply - UL Listed
This is an FCC/CE certified and UL listed power supply. Need a lot of 5V power? This switching supply gives a clean regulated 5V output at up to 2000mA. 110 or 240 input, so it works...
https://www.adafruit.com/product/276

USB cable - USB A to Micro-B
This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...
https://www.adafruit.com/product/592

Premium Male/Male Jumper Wires - 20 x 3" (75mm)
Handy for making wire harnesses or jumpering between headers on PCB's. These premium jumper wires are 3" (75mm) long and come in a 'strip' of 20 (2 pieces of each...
https://www.adafruit.com/product/1956

JST-PH Battery Extension Cable - 500mm
By popular demand, we now have a handy extension cord for all of our JST PH-terminated battery packs (such as our Lithium/LiPoly and 3xAAA holders). One end has a JST-PH compatible...
https://www.adafruit.com/product/1131
Hook-up Wire Spool Set - 22AWG Solid Core - 6 x 25 ft
Perfect for bread-boarding, free wiring, etc. This box contains 6 spools of solid-core wire. The wire is easy to solder to and when bent it keeps its shape pretty well. We like to have...
https://www.adafruit.com/product/1311

Female DC Power adapter - 2.1mm jack to screw terminal block
If you need to connect a DC power wall wart to a board that doesn't have a DC jack - this adapter will come in very handy! There is a 2.1mm DC jack on one end, and a screw terminal...
https://www.adafruit.com/product/368

Hakko Professional Quality 20-30 AWG Wire Strippers
These are the finest wire strippers we have used, and if you have to do a lot of wiring, you will agree! They have soft rounded grips - very comfortable to use, and precision ground...
https://www.adafruit.com/product/527

Materials

- Xylophone ()
- Wire strippers
- Laser cutter and acrylic (optional)

Doing some prep work ahead of time can simplify this build immensely. Trimming and setting aside the wires you'll need can save yourself confusion later on. Follow this image to find and set aside what you'll need before diving into building the circuit.
Play On

Read on to learn how to build your own robotic musician!
Solenoid Circuit

If you'd like to breadboard the circuit before soldering it together permanently, follow this fritzing diagram. (Note the picture below shows the Metro, the Grand Central wiring is identical)

If you'd like to jump into soldering, scroll down to see the steps.

Gather what you'll need
The N-channel Power MOSFET

Because solenoids draw a lot of current at once, each will be connected to a MOSFET transistor. From left to right, the legs of this MOSFET are Gate (G), Drain (D), and Source (S).

Cut and strip 8 red and 8 black each about 1 inch in length

Cut eight pairs of blue wires of varying lengths. These will be used to connect each solenoid unit to its digital pin on the board.
Eight pairs of short male-to-male jumper wires will be used to create robust connections to each individual solenoid.

Set aside eight N-channel power MOSFET transistors and eight 1N4001 diodes.

The N channel MOSFETs will handle switching power on and off to the solenoids. These FETs can switch an impressive 60A and 30V and are in TO-220 packages so they fit nicely into the proto-board.

The classic 1N4001 power blocking diode will be used here for kickback protection for our solenoids.
Soldering the Circuit

Solder each of the 8 MOSFETs in place.

You will notice that 8 MOSFETs overrun the laterally connected pins on the proto board. This is ok, we will create our own connections where needed later.
Solder in the 1N4001 diodes.

Be careful of orientation!

Clip off the excess leads, but be careful not to clip the leads further down! You will need these to make connections later.
Solder in the red wires connecting the power line to the circuit.

Once again, clip off leads where necessary, leaving extra where the pads aren't already connected.
Time for the ground connections!

Use these wires to connect the ground line to the Source (S) leg of each MOSFET.
Using the blue wires, connect Gate (G) leg of the MOSFET to pins 2 through 9 on the Grand Central.

Use the wire strippers to size your wires to the lengths you need.
Now, importantly, we must connect the ground line of this circuit to Grand Central's ground. Don't forget this step!
Solder the jumper wires in place, connecting one wire to each end of the 1N4001 diodes.

As before, trim excess where necessary.
Now it's time to fold over the extra length of wire we've left sticking through the bottom and make the missing connections where the pads aren't already connected.
The Plug

Connect a red and black wire to the female DC terminal block adapter.

Solder these wires to the ground and power lines on the proto board (labeled "G" and "+").
Connect JST Extension Cables

These male-male jumper wires make reliable connections between the proto board and JST cables, as well as making your bundle of wires easy to organize.
Add Headers

Finally, solder headers onto your proto-shield to connect it to the Grand Central.

Insert the headers into your board (you don't need to add the long 2x20 header for this project).

Place your assembled proto-shield on top of the headers.

Solder in place!
Time to move on to the next step!

Assemble the Frame

You will need some structure to position the solenoids above the xylophone.

The .AI file below contains a design suited for laser cutting out of 1/8" thick acrylic. If you don't have access to a laser cutter, you can use this template to cut one out of another material such as wood or cardstock, or design something new for your particular xylophone situation.

Xylophone-frame.ai
Some small 0-80, 1/4" thread screws () are perfectly sized to hold each solenoid to the bar.

A tiny screwdriver () is useful in this instance.

Height Adjustable

You will notice there is some wiggle room in the slots where the top bar fits in. This allows the user to precisely position the bar above the xylophone so each solenoid hits the note perfectly.
CircuitPython

Getting Familiar

CircuitPython is a programming language based on Python, one of the fastest growing programming languages in the world. It is specifically designed to simplify experimenting and learning to code on low-cost microcontroller boards.

CircuitPython is easiest to use within the Mu Editor. If you haven't previously used Mu, [this guide will get you started](https://circuitpython.readthedocs.io/en/latest/). If you haven't used CircuitPython together before, make sure you've updated it with the latest version of CircuitPython. [This guide will show you how](https://circuitpython.readthedocs.io/en/latest/).
Demo Code

Plug your Grand Central into your computer (mac/PC/Linux) via a known good USB A to micro-B cable. Your board should appear to the computer as a flask disk drive named CIRCUITPY. If you see a disk name GCM4BOOT, try to press the reset button again. If the only drive name you get is GCM4BOOT, CircuitPython may not be loaded on the board. You can load CircuitPython per this guide.

Copy code.py from the link below and put it in CIRCUITPY root directory. You can work with this code in any text editing application, or open and save with Mu() if you prefer.

```python
# SPDX-FileCopyrightText: 2019 Dano Wall for Adafruit Industries
# SPDX-FileCopyrightText: 2019 Anne Barela for Adafruit Industries
# # SPDX-License-Identifier: MIT
# Adafruit Grand Central Robot Xylophone Demo Program
# Dano Wall and Anne Barela for Adafruit Industries
# MIT License

import time
import board
from digitalio import DigitalInOut, Direction

solenoid_count = 8  # Set the total number of solenoids used
start_pin = 2       # Start at pin D2

# Create the input objects list for solenoids
solenoid = []
for k in range(start_pin, solenoid_count + start_pin + 1):
    # get pin # attribute, use string formatting
    this_solenoid = DigitalInOut(getattr(board, "D{}".format(k)))
    this_solenoid.direction = Direction.OUTPUT
    solenoid.append(this_solenoid)

STRIKE_TIME = 0.01  # Time between initiating a strike and turning it off
TIME_BETWEEN = 0.5  # Time between actions in seconds

song = [3, 4, 5, 4, 3, 3, 4, 4, 4, 3, 3, 4, 5, 4, 3, 3, 2, 2, 3, 4, 5]

def play(key, time_to_strike):
    solenoid[key].value = True
    time.sleep(time_to_strike)
    solenoid[key].value = False

def rest(time_to_wait):
    time.sleep(time_to_wait)

while True:
    # Play each of the bars
    for bar in range(solenoid_count):
        play(bar, STRIKE_TIME)
        rest(TIME_BETWEEN)

    time.sleep(1.0)  # Wait a bit before playing the song

    # Play the notes defined in song
    # simple example does not vary time between notes
    for bar in range(len(song)):
```

©Adafruit Industries
play(song[bar], STRIKE_TIME)
rest(TIME_BETWEEN)
time.sleep(1.0)

The demonstration shows a simple scale and a short tune. You can use the play function to strike the various tones for a specified length of time then rest between notes. This allows you to do more complex melodies.

Have fun with your new musical instrument!

Troubleshooting

Problem: Grand Central M4 isn't recognized by Mu!

Solution: Make sure your board is set up with CircuitPython, which has the board show up as a flash drive named CIRCUITPY when you connect it to your computer. If it is showing up as GCM4BOOT on your computer, you can follow the steps in this guide to ensure CircuitPython is loaded and you see the CIRCUITPY drive.