MIDI Solenoid Drummer

Created by Collin Cunningham

https://learn.adafruit.com/midi-solenoid-drummer

Last updated on 2024-07-15 03:55:17 PM EDT

Table of Contents

Overview

This guide will show you how to build miniature percussion instrument which can be controlled via MIDI over USB. The example project is intended to serve as a jumping-off point for your own designs. Read about how I built mine, then experiment to make a unique instrument you can control with your computer.

Parts

1 x Feather M0 Basic Proto — https://www.adafruit.com/product/2772
microcontroller to run project code

1 x Crickit Featherwing — https://www.adafruit.com/product/3343
allows Feather to control solenoids

1 x 5V 2A Power Supply — https://www.adafruit.com/product/276
provides power for the Crickit board

4 x 5V Solenoid — https://www.adafruit.com/product/2776
strikes the drums!

1 x JST-PH Battery Extension Cable - 500mm — https://www.adafruit.com/product/1131
connects solenoids to the Crickit board

1 x USB Micro Cable — https://www.adafruit.com/product/2185
for programming the Feather

1 x Small phillips head screwdriver — https://www.adafruit.com/product/3284
for securing wires to Crickit's screw terminals

connects Crickit power to solenoids

1 x 5-wire Block Connector https://www.adafruit.com/product/874

connects Crickit power to solenoids

---

1 x Mounting board

Something to mount your drumkit on

---

8 x #4 screws

for mounting solenoids

---

8 x #4 washers

for mounting solenoids

---

4 x DIY drums

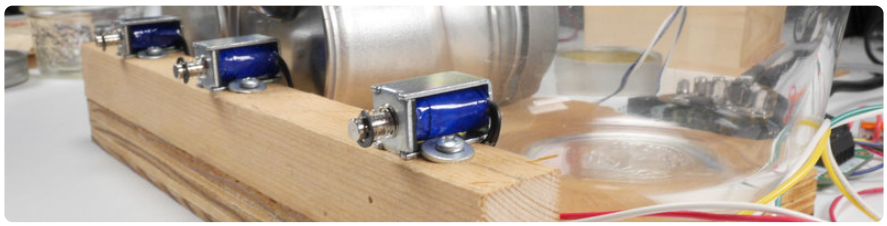
any small objects that make cool sounds when struck

---

1 x Hot glue gun and glue sticks

for mounting drums

Wiring

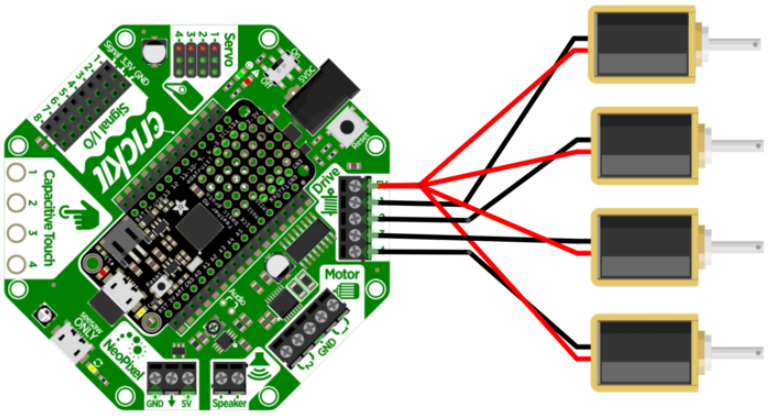


Wiring the drummer is relatively easy. **Feather** connects to the **Crickit** board, and each **solenoid** connects to its own **Crickit Drive port**. Let's look take a closer look ...

Connect the Feather & Crickit

Solder the included male header pins (https://adafru.it/u3F) to your **Feather** board and attach it to the **Crickit** via the central **female headers**.

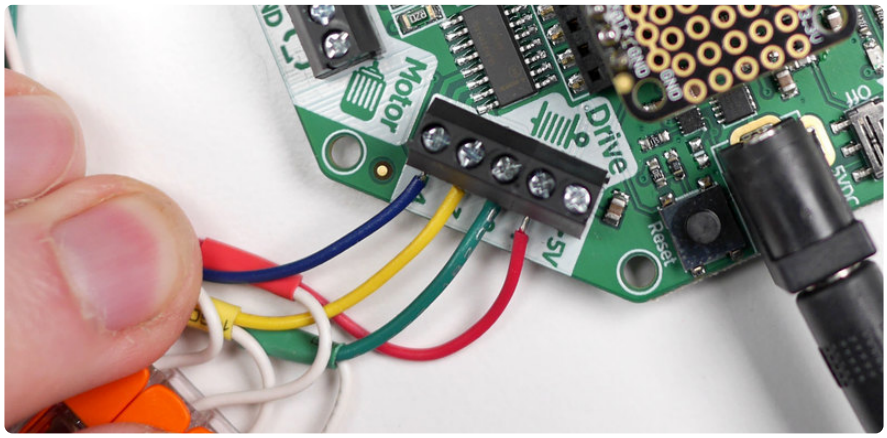
Connect solenoids



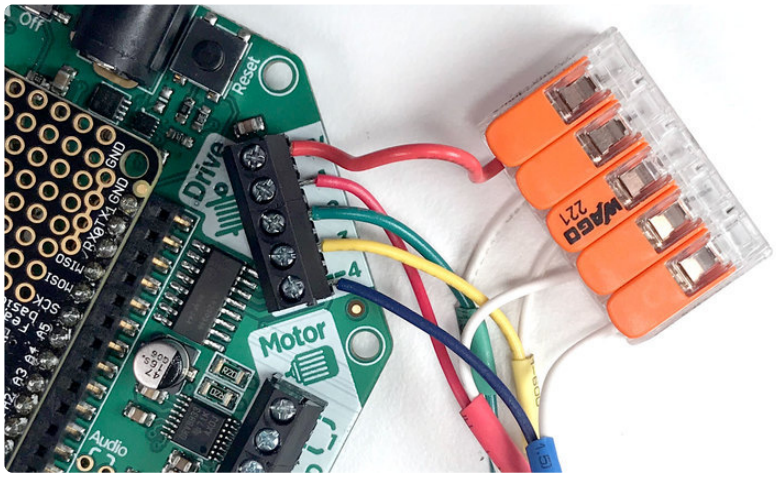
Each **solenoid** has two wires which need to be connected to the **Crickit's Drive section**. One of them needs to connect to the **5V port**, and the other connects to one of the **numbered Drive ports**.

Note that the wires in the above diagram are **colored for clarity**. The actual solenoid wires are **both black**, indicating that can be **connected either way**.

The **solenoid wires** are short, so we'll need to **extend them** in order to reach a convenient **Crickit** location. I made my own extension cables with stranded wire and small male machine pins, but it's much easier to use JST cables (http://adafru.it/3814). Just **clip the female connectors** off the JST cable and you've got a 500mm extension.

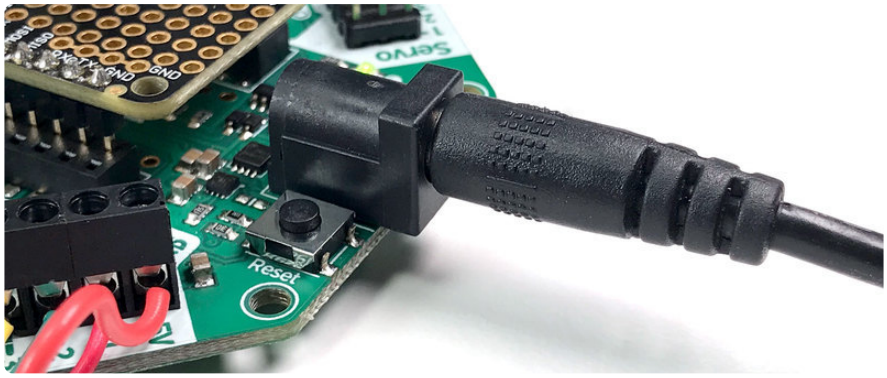


Connect one wire from each **solenoid** to one of the **Crickit's Drive ports** as seen above. Use a small **phillips head screwdriver** to secure each wire in place.



Next, you'll need to connect the **four remaining solenoid wires** to **Crickit's 5V port**. I used a small piece of wire to connect the **5V port** to a 5-wire block connector (http://adafru.it/874) - this turns **one port into four**. Alternatively - you could use a terminal block (http://adafru.it/677).

Connect power



Finally, connect the **Crickit** to the **5V wall supply,** turn the **Crickit on**, and you're good to go.

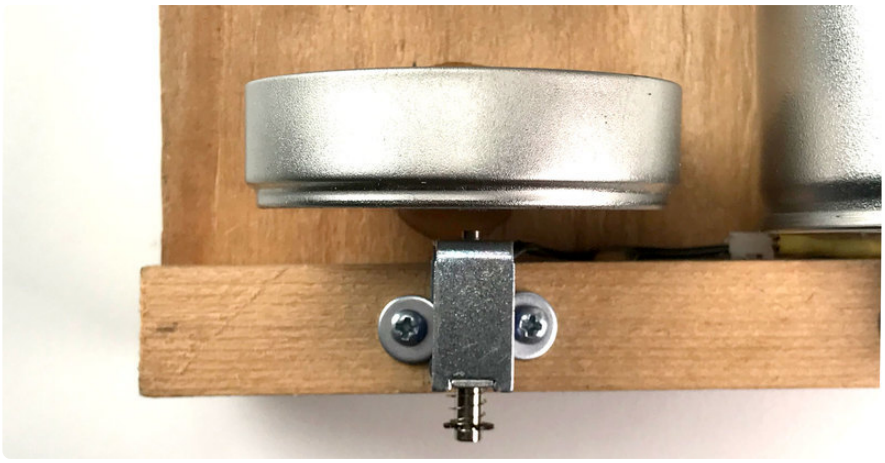
Design your drumkit

After some experimenting, I found a few objects which made relatively interesting percussion sounds when tapped - namely:

- Matcha cans
- Small Pyrex bowl
- Plastic shot glass + beads

The **Pyrex bowl** was ready to use a **cymbal** without any modification, but the matcha can and shot glass shaker would need a little creative engineering.

To get a lower tone from my **matcha can**, I attached an adhesive rubber bumpon/foot (http://adafru.it/550) to the **bottom surface**. I noticed I could lower the sound even further by removing the can's lid and positioning it a few millimeters away from the can itself. It's no 808 kick, but it will serve as a relatively low tone in my setup.



To complement the can **kick drum**, I'll also use an extra can lid as a standalone **stick/click drum**.

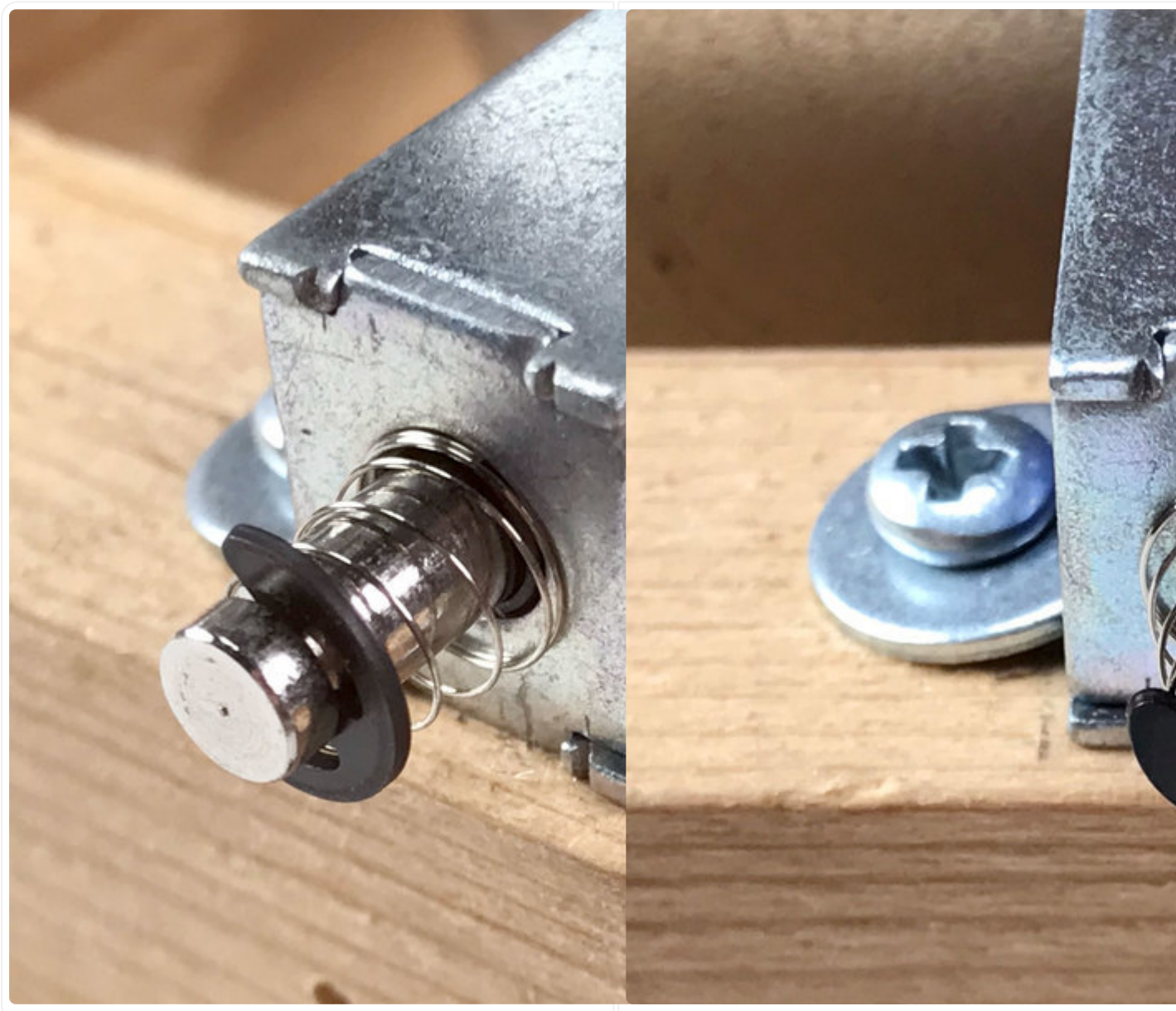
For a **shaker**, I used a clear **plastic shot glass** and poured in some **plastic beads**. A small amount of dry rice/couscous/candy bits could work too, just be sure whatever shaking medium you use is light. The small 5V solenoid will struggle to push heavier objects.

Platform

To hold all the drums and solenoids in place, I pulled a piece of **9"x11" scrap wood** from my junk bin. The solenoids needed to be raised up a bit from the board in order to hit the sweet spots on my drums - a **3/4" square dowel** works well for this. I cut the

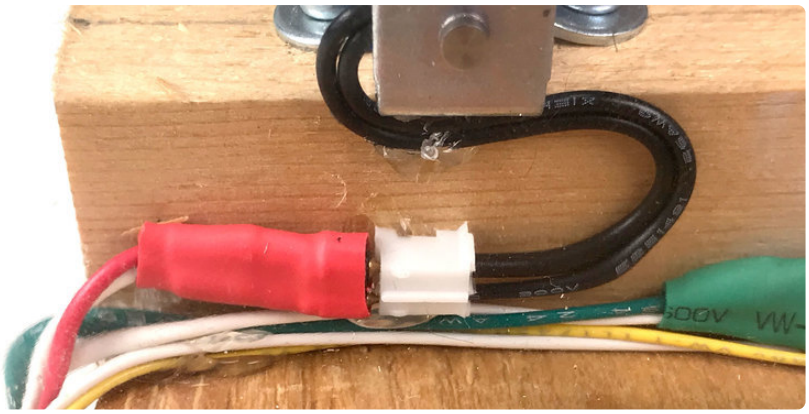
dowel to match the length of the board and glued it down, leaving it clamped overnight to cure.

Mounting the solenoids



Each solenoid is held in place using **screws & washers** - which makes repositioning and small adjustments easy. I marked the ideal position for each solenoid, and **partially** screwed in the scew/washers on each side (~18mm apart) of the solenoids future home. I then slid the solenoids in between the screws and secured them in place so the **washers act as clamps**.

For a different take on solenoid mounting, check out the LEGO compatible mounts in this episode of John Park's Workshop (https://adafru.it/Cu3).



To keep things tidy - all solenoid wires are held in place with hot glue. Hot glue is a good thing.

Be careful using hot glue - the glue gun tip and the glue itself may cause burns when hot.

Mounting the drums

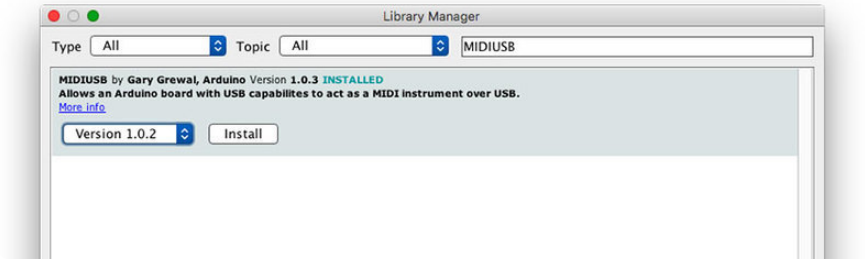
I experimented with different drum positions by firing a solenoid repeatedly and holding the drum at different distances. Once I found a good spot, I secured the drum in place with generous amounts of **hot glue**.

I attached the **shaker** directly to the solenoid's piston/pin with **hot glue**. The weight of the shot glass & beads can slow down the solenoid if the glass drags against the wood too much - **adding a single washer** below the shaker helped to ensure it moved smoothly.

Code

We'll use the Arduino IDE for the drummer's code - download & install it (https://adafru.it/Cto) if you haven't already.

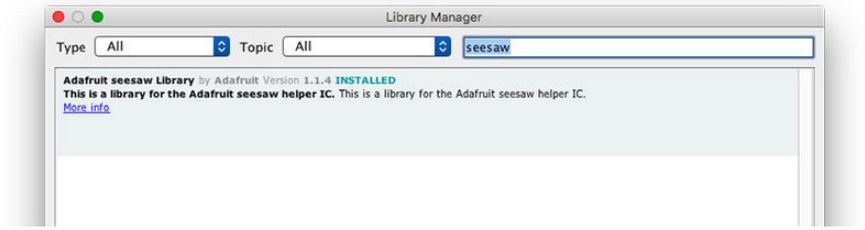
If you're new to Arduino, check out the Getting Started with Arduino guide (https://adafru.it/dMN).



Along with Arduino IDE, we'll need to install the following two code libraries (https://adafru.it/aYM).

Connect your Feather to your computer and open up the Arduino IDE. From the top menu, go to **Sketch --> Include Library --> Manage Library**, search for **MIDIUSB** and install the latest version of the **MIDIUSB** library.

Next follow the **same steps as above** to install the **Seesaw** library ...



Additionally, go to **Tools --> Board: --> Board Manager** and update the **Adafruit SAMD boards** library if you haven't already. Additional info for installing boards in the Arduino IDE is available here (https://adafru.it/BAV).

Upload code

Go to **Tools --> Board**, and choose **Adafruit Feather M0**. Then go to **Tools --> Port** and choose the corresponding port for your board.

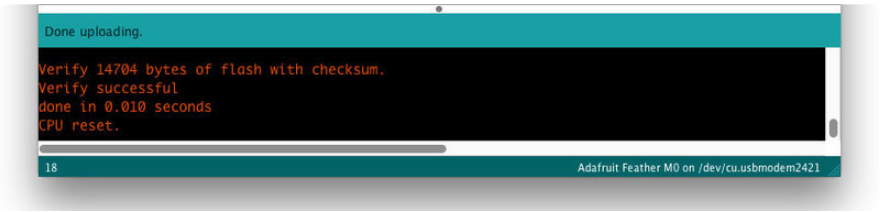
Create a **new sketch**, **copy the code** you see below, and **paste** it into that new sketch.

```
// SPDX-FileCopyrightText: 2018 Collin Cunningham for Adafruit Industries
//
// SPDX-License-Identifier: MIT

/*  MIDI Solenoid Drummer
 *  for use with Adafruit Feather + Crickit Featherwing
 *  assumes a 5V solenoid connected to each of Crickit's four Drive ports
 */

#include "Adafruit_Crickit.h"
#include "MIDIUSB.h"
```

```
Adafruit_Crickit crickit;

#define NUM_DRIVES 4
int drives[] = {CRICKIT_DRIVE1, CRICKIT_DRIVE2, CRICKIT_DRIVE3, CRICKIT_DRIVE4};
int cym = CRICKIT_DRIVE4;
int kick = CRICKIT_DRIVE3;
int snare = CRICKIT_DRIVE2;
int shake = CRICKIT_DRIVE1;
int hitDur = 8; //solenoid on duration for each hit (in milliseconds)

void setup() {

  if (!crickit.begin()) {
    while (1);
  }

  for (int i = 0; i < NUM_DRIVES; i++)
    crickit.setPWMFreq(drives[i], 1000);  //default frequency is 1khz

  test(); //test solenoids at start
}

void loop() {

  midiEventPacket_t rx = MidiUSB.read();  //listen for new MIDI messages

  switch (rx.header) {
    case 0x9:            //Note On message
      handleNoteOn(
        rx.byte1 & 0xF,  //channel
        rx.byte2,        //pitch
        rx.byte3         //velocity
      );
      break;
    default:
      break;
  }
}

void handleNoteOn(byte channel, byte pitch, byte velocity) {

  switch (pitch) {
    case 24:      //kick = C1/24
      hit(kick);
      break;
    case 25:      //snare = C#1/25
      hit(snare);
      break;
    case 26:      //shake = D1/26
      hit(shake);
      break;
    case 27:      //cymbal = D#1/27
      hit(cym);
      break;
    default:
      break;
  }
}

void hit(int drum) {
  crickit.analogWrite(drum, CRICKIT_DUTY_CYCLE_MAX);  //turn solenoid all the way on
  delay(hitDur);                                      // wait
  crickit.analogWrite(drum, CRICKIT_DUTY_CYCLE_OFF);  //turn solenoid all the way
off
}

void test() {   //for debugging
  hit(cym);
  delay(400);
```

```
  hit(kick);
  delay(400);
  hit(snare);
  delay(400);
  hit(shake);
  delay(400);
}
```

Click the **Upload** button and wait for the process to complete. Once you see **Done Uploading** at the bottom of the window, the Feather should automatically fire each solenoid to **test your drumkit**.



Usage

To send MIDI to the drummer from your computer's USB port, you'll need a MIDI-capable music application such as Garageband (https://adafru.it/Ctp), Reaper (https://adafru.it/Ctq), or Ableton Live (https://adafru.it/Ctr). You can use a **MIDI keyboard** with your music app to control the the drums, but it's more fun to **compose sequences** and play them back over MIDI. Fast, complex sequences actually sound pretty impressive on this little kit.

Here's my kit playing a sequence from Ableton Live ...

MIDI Map

The code responds to MIDI **note on** messages for the following **note/pitch** numbers:

- 24 (C1) = port 2
- 25 (C♯1) = port 3
- 26 (D1) = port 4
- 27 (D♯1) = port 1

Solenoids will be triggered by these notes on **any MIDI channel**.

Limitations

Because the drummer code needs to pause and wait for a very short amount of time while each solenoid is powered on, it is only able to process one note at a time. This means that when 2 different drum notes are received in fast succession, only one will be played.