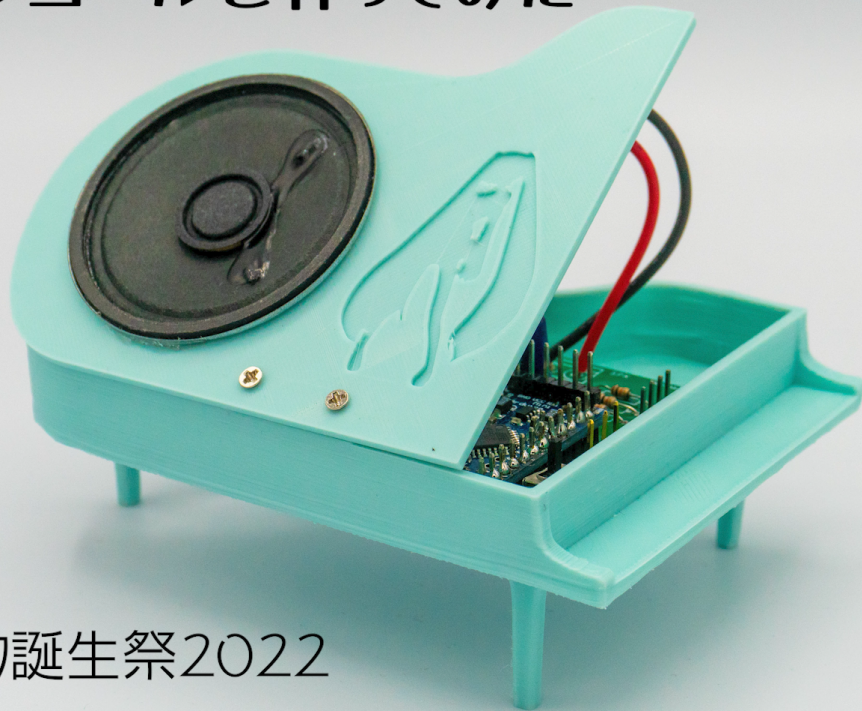


# 我使用YMZ294制作了Famicom风格的电子音乐盒[初音未来生日祭2022]

电子工作

電子オルゴールを作ってみた



#初音ミク誕生祭2022

© 2022年8月31日

## 背景

我是在收到电子零件分销商 Mouser Electronics 的提议后制作的，希望我能制作一些与 8 月 31 日初音未来生日庆典相关的电子作品。我们使用 Mouser 网站上名为 Forte 的零件列表创建工具来选择零件，并使用了 Mouser 提供的一些零件。

- 电容100uF
- 振荡器4MHz
- Arduino Pro 迷你版

顺便说一句，我在Mouser的网站上找不到用作音源IC的YMZ294（因为它是旧IC），所以我从Akizuki Denshi 购买了它。（您也可以在 Aliexpress 上购买，但您会收到一个您不知道的兼容商品！）

<https://akizukidenshi.com/catalog/g/gI-12141/>

## 电影

## YouTube

電子オルゴールを作ってみた【初音ミク誕生祭2022】



## 妮可・妮可・杜加

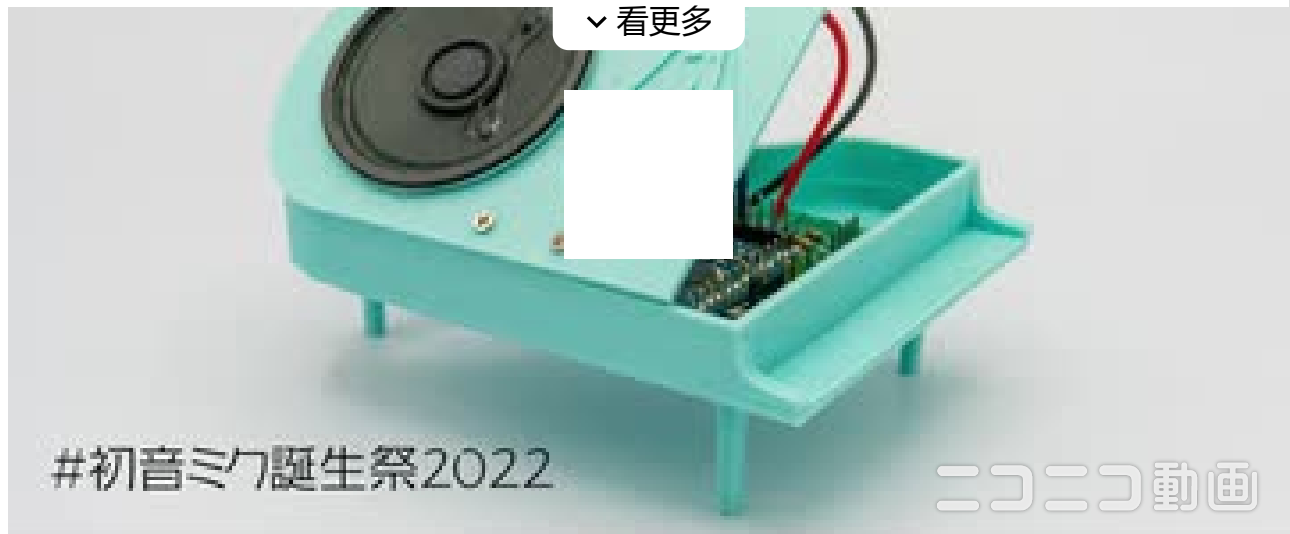


電子オルゴールを作ってみた【初音ミク誕生祭202...



作ってみた 初音ミク誕生祭2022 初音ミク YMZ294

▼ 看更多

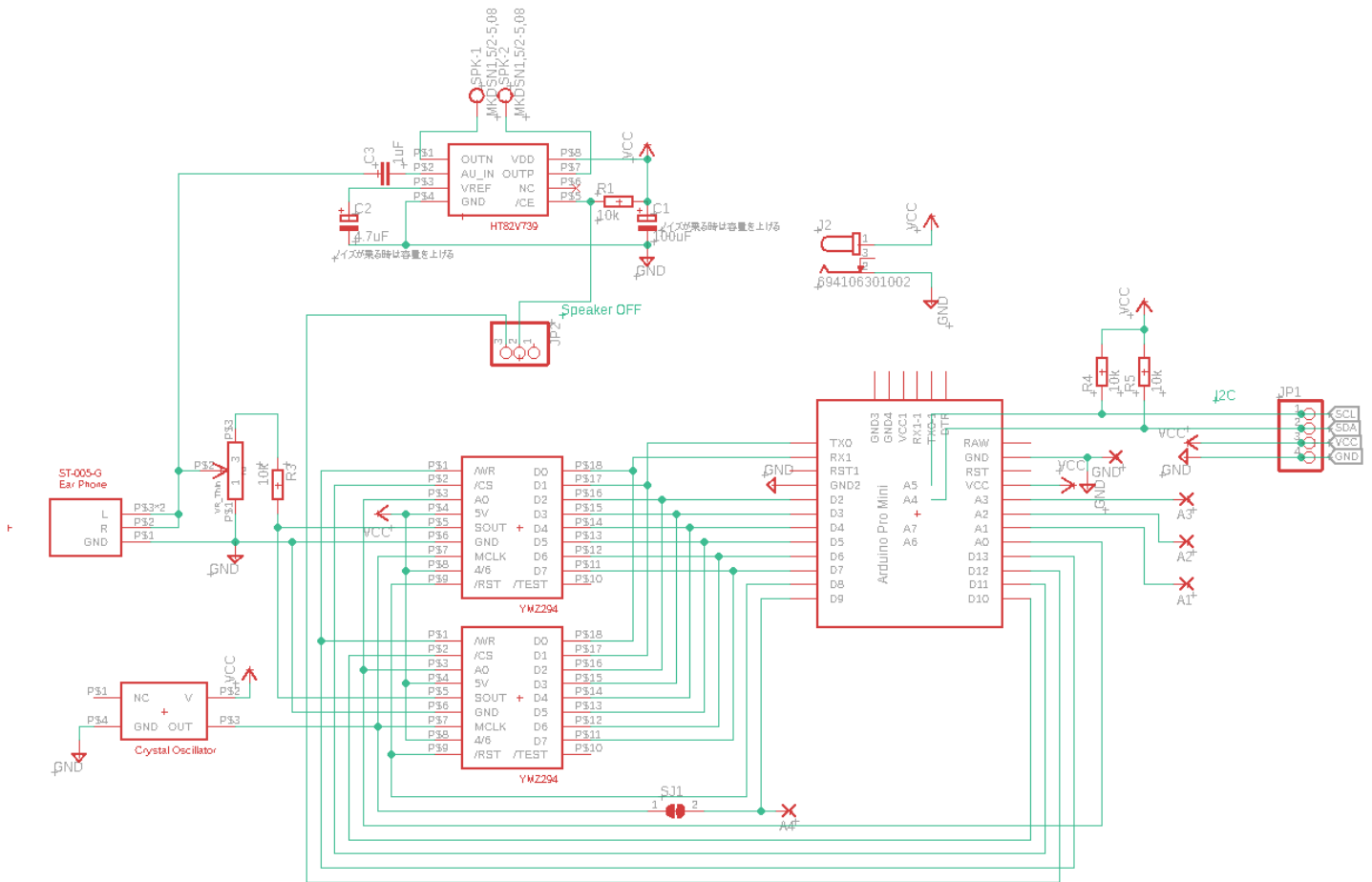


## 解释

声音使用雅马哈音源IC YMZ294 产生。这是一个相对简单的 IC，可以像 Famicom 一样演奏最多 3 个和弦，而该板使用其中两个 IC 最多可以演奏 6 个和弦。该IC的输出不是模拟输出，而是似乎改变了数字矩形波的周期，因此看来输出可以按原样连接以合成3和弦+3和弦。

我不擅长模拟电路，但这一次声音相对简单。我用可变电阻（VR\_Thin）改变音量，并使用名为 HT82V739 的放大器 IC 作为扬声器。该放大器不仅可以通过微控制器打开/关闭，还可以通过手动开关（跳线：JP2）打开/关闭。这是一个当您不想让扬声器播放时可以关闭的功能（您可以通过耳机收听）。另一个特点是YMZ294需要一个4MHz振荡器，这是一种输出一定频率的元件，但秋月电子不出售它。当然，您可以从 Mouser 购买它（是的），但由于 Arduino Pro Mini 的工作频率为 8MHz/16MHz，因此即使没有振荡器，我也可以通过将此频率发送到 YMZ294 来发送 D9 引脚信号。（短接 SJ1）。该电路的失败之处在于我使用Arduino Pro Mini的

SPI输出 (DP10、11、12、13) 来控制YMZ294。如果我使用SPI输出读取SD卡, 我可以支持长时间播放.....  
失败的原因是最初尝试通过从另一台设备发送 I2C 输入进行播放。



## Arduino代码

### Arduino代码体

track0 是 "Melt" 的 MIDI 代码。另外, 与上面的接线不同, 连接 AD\_PIN 和 A0 的部分实际上连接到 A3, 因此已在代码中更正。(显然使用的Arduino Pro Mini库与实际不同)

```

1  #include <YMZ294Kai.h> //https://wp.hrmux.com/?page_id=991
2  //YMZ294Kai.hはCSとWRを別のピンで指定するライブラリ。複数のYMZ294を繋ぐ用
3  #include <avr/pgmspace.h>
4
5  #include <SoftwareSerial.h>
6  SoftwareSerial mySerial(A2, A1); // RX, TX
7
8  // Output Pins
9  const byte WR_PIN = 13;
10 const byte CS1_PIN = 11;
11 const byte CS2_PIN = 10;
12 const byte AD_PIN = A3;
13 const byte RESET_PIN = 8;
14 const byte CLKOUT_PIN = 9;
15 const byte AmpEn_PIN = 12;
16
17 YMZ294 ymz1(WR_PIN, CS1_PIN, AD_PIN, RESET_PIN);
18 YMZ294 ymz2(WR_PIN, CS2_PIN, AD_PIN, RESET_PIN);
19
20 //MIDI Setting
21 const byte ch = 6;

```

<https://prototype09.com/archives/589>

```
91     readPos++;
92     byte c = pgm_read_byte(readPos);
93     switch(c){
94         case 0x80://Note Off
95             readPos++;
96             c = pgm_read_byte(readPos);//音程
97             //mySerial.print("tone OFF:");
98             //mySerial.print(c ,HEX);
99             for(int i=0;i<ch;i++){
100                 if(tones[i] == c){
101                     readPos++;
102                     byte v = pgm_read_byte(readPos);//ボリューム
103                     if(i<3){
104                         ymz1.SetFreqVol(i, 0, 0);
105                     }
106                     else{
107                         ymz2.SetFreqVol(i-3, 0, 0);
108                     }
109                     tones[i] = 0x00;
110                     //mySerial.print(",CH");
111                     //mySerial.print(i);
112                     //mySerial.print(",Volume:");
113                     //mySerial.println(v);
114                     break;
115                 }
116             }
117             //LED OFF
118             digitalWrite(13,LOW);
119             break;
120         case 0x90:
121             readPos++;
122             c = pgm_read_byte(readPos);//音程
123             //mySerial.print("tone ON:");
124             //mySerial.print(c ,HEX);
125             for(int i=0;i<ch;i++){
126                 if(tones[i] == 0x00){
127                     readPos++;
128                     byte v = pgm_read_byte(readPos);//ボリューム
129                     if(i<3){
130                         ymz1.SetFreqVol(i, noteFreq[c], (v>>3));
131                     }
132                     else{
133                         ymz2.SetFreqVol(i-3, noteFreq[c], (v>>3));
134                     }
135                     tones[i] = c;
136                     //mySerial.print(",CH");
137                     //mySerial.print(i);
138                     //mySerial.print(",Volume:");
139                     //mySerial.println(v);
140                     break;
141                 }
142             }
143             break;
144         case 0xff:
145             readPos++;
146             c = pgm_read_byte(readPos);
147             if(c == 0x2f){
148                 enable = false;
149             }
150             break;
151     }
152     //tick読み込み
153     readPos++;
154     readTick();
155     //mySerial.println(micros()-t);
156 }
157
158 if(tick>0){
159     while(t+tempoDelay*tick>=micros()){}
```

```
160     tick = 0;
161   }
162 }
163 }
164
165 void readTick(){
166   tick = 0;
167   uint32_t t = pgm_read_byte(readPos);
168   //mySerial.print("t:");
169   //mySerial.println(t, HEX);
170   while(t>=0x80){
171     readPos++;
172     tick = (tick<<7) + t - 0x80;
173     t = pgm_read_byte(readPos);
174     //mySerial.print("t:");
175     //mySerial.println(t, HEX);
176   }
177   tick = (tick<<7) + t;
178   //mySerial.print("tick:");
179   //mySerial.println(tick, HEX);
180 }
```

## YMZ294凯库

使用的原始库是在[https://wp.hrmux.com/?page\\_id=991](https://wp.hrmux.com/?page_id=991)上发布的库。为了能够控制多台YMZ294，原库中同时控制WR和CS的部分是分开使用的。"YMZ294 ymz1(WR\_PIN, CS1\_PIN, AD\_PIN, RESET\_PIN);" 定义YMZ294 时，如果将 WR\_PIN 和 CS1\_PIN 设置为相同的引脚号，则可以像原库一样使用，但这样的话，YMZ294 就只能一部分。我只能用它。

[YMZ294Kai.zip](#)

## 今后

这次正好赶上了初音未来的生日祭，做了一个简单的电子音乐盒，但本来我想做一个可以用摄像头读取代码并播放的设备，所以我决定考虑一下。我也想继续。