MAGIC WITH PLAY

Sensor

MIDI

Max/MSP

Music Apps

Interactive Electro-acoustic Music

遇见声音

Sound Encounters

母体之内

发.声

耳.语

波的形状

Interactive Electro-acoustic Music

Sound Encounters

母体之内

第一次聆听

Interactive Electro-acoustic Music

Sound Encounters

发.声

声音的本质是振动

发声体在外力的作用下围绕其初始位置做往返运动

弹簧

Interactive Electro-acoustic Music

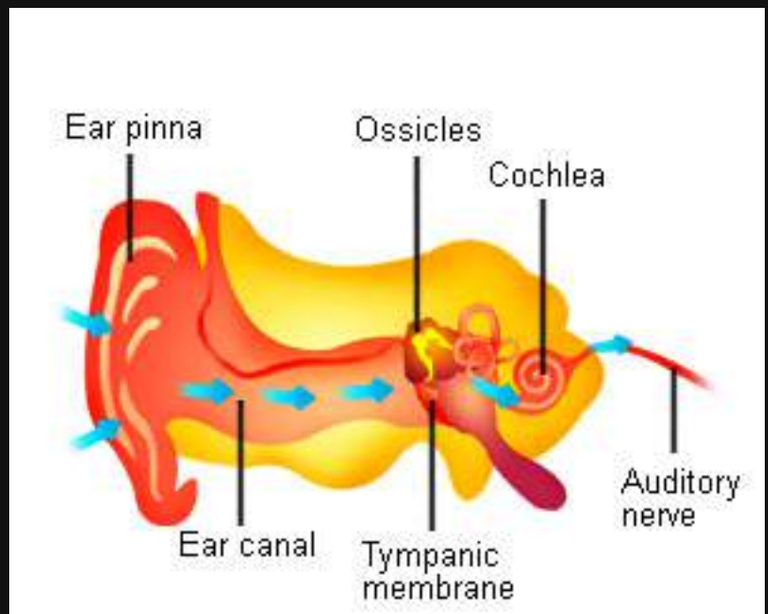
Sound Encounters

耳.语

人耳如何听到声音?

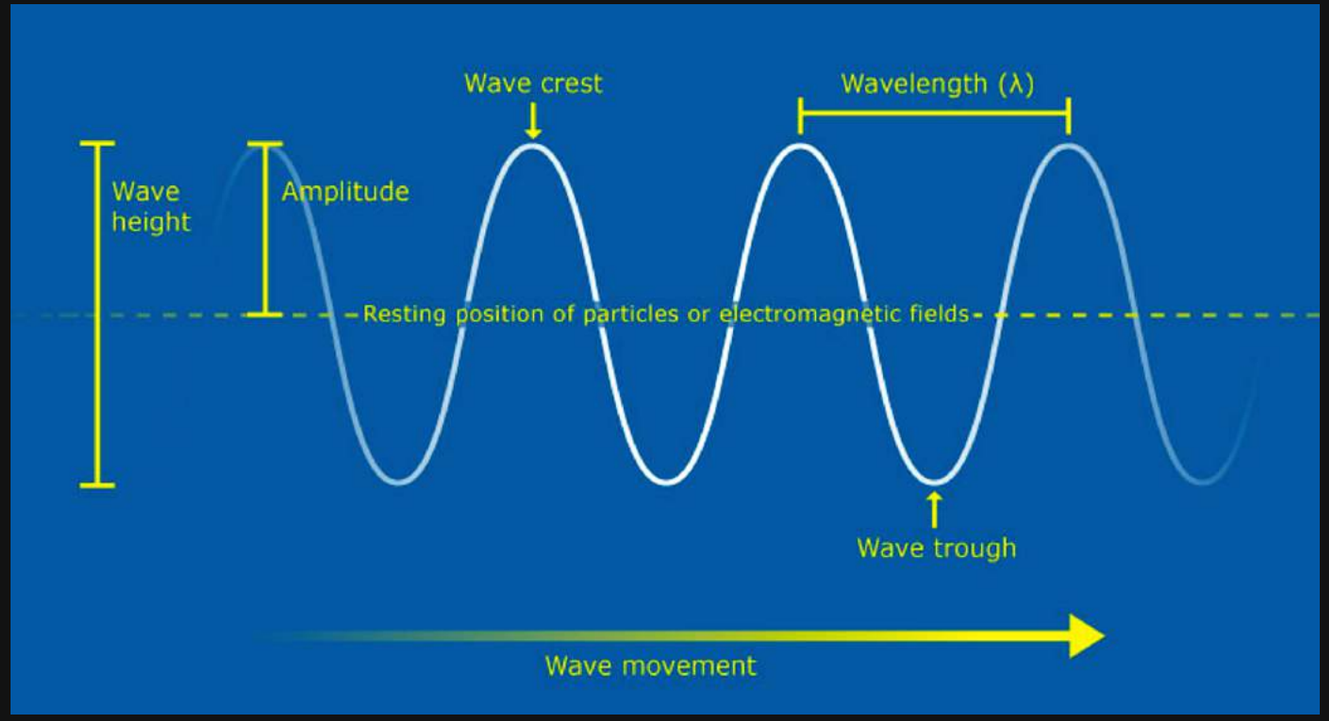
振动源 媒介 鼓膜 听觉神经 大脑

20 赫兹 - 20000 赫兹



波的形状

简谐振动曲线
Pure sine wave



频率 Frequency

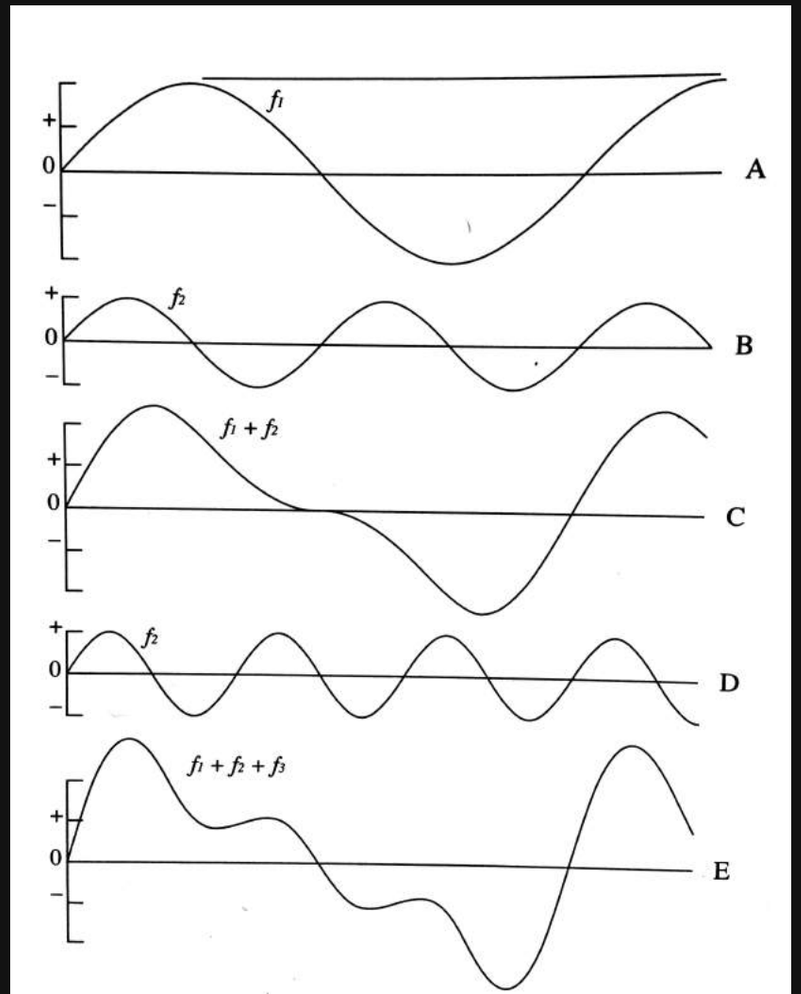
1 秒内振动的次数，用赫兹 Hz 表示

振幅 Amplitude

物体振动的幅度，用分贝 dB 表示

波的形状

非理想态的声音



现实中听到的绝大多数声音包含多个频率

任何声音都可以被分解为具有不同频率、振幅和相位的正弦波的叠加

人声：瞬态复合声

Interactive Electro-acoustic Music

以声为乐

Sound to Music

录.音

一条音轨

Interactive Electro-acoustic Music

Sound to Music

录.音

记录的是声波

麦克风的工作原理

声音的振动传到麦克风的振膜，振动推动磁铁形成电流，

再通过电路放大电信号

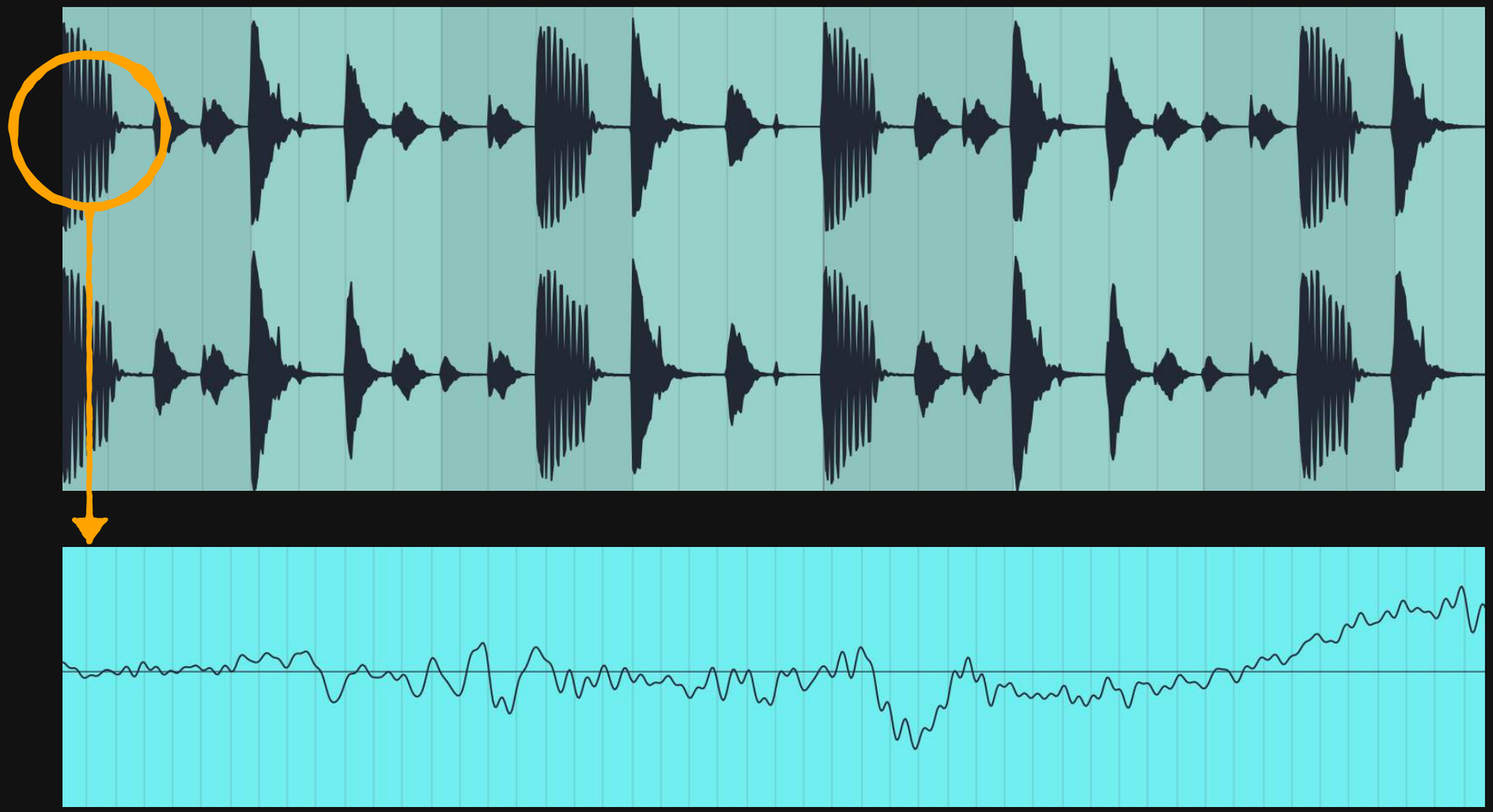
接触式麦克

压电蜂鸣片 3.5 音频线

Interactive Electro-acoustic Music

Sound to Music

一条音轨



Interactive Electro-acoustic Music

学啥呦

Teaching

已.知

Play with Magic

Hack without Acoustic

Interactive Electro-acoustic Music

Teaching

已.知

Arduino

Sensor

压力 声音 红外 光敏 超声波 弯曲 触摸 三轴加速度 心率 颜色 温度 湿度

薄膜压力 水流速度 碰撞 线性滑动 灰度

Motor

Mechanical

WIFI

Fabrication

Interactive Electro-acoustic Music

Teaching

Play with Magic

MIDI & Music Apps

GarageBand / Logic Pro / Ableton Live

Max/MSP

Hybrid Visual Programming System

破解版软件豪华大礼包

Interactive Electro-acoustic Music

Teaching

Hack without Acoustic

聆听电磁场

Contact mike / Pick up / Speaker / Mini Amplifier

模拟声场

Oscillator / Amplifier / Shifting / Filtering / Power Amp

集成录放

Arduino MP3 Shield

声音探测

Audio Analysier / Audio Recognition

Interactive Electro-acoustic Music

MIDI

Music Instrument Digital Interface

协.议

格式化

控制器

发.射

虚匿语气

Apps

Interactive Electro-acoustic Music

MIDI

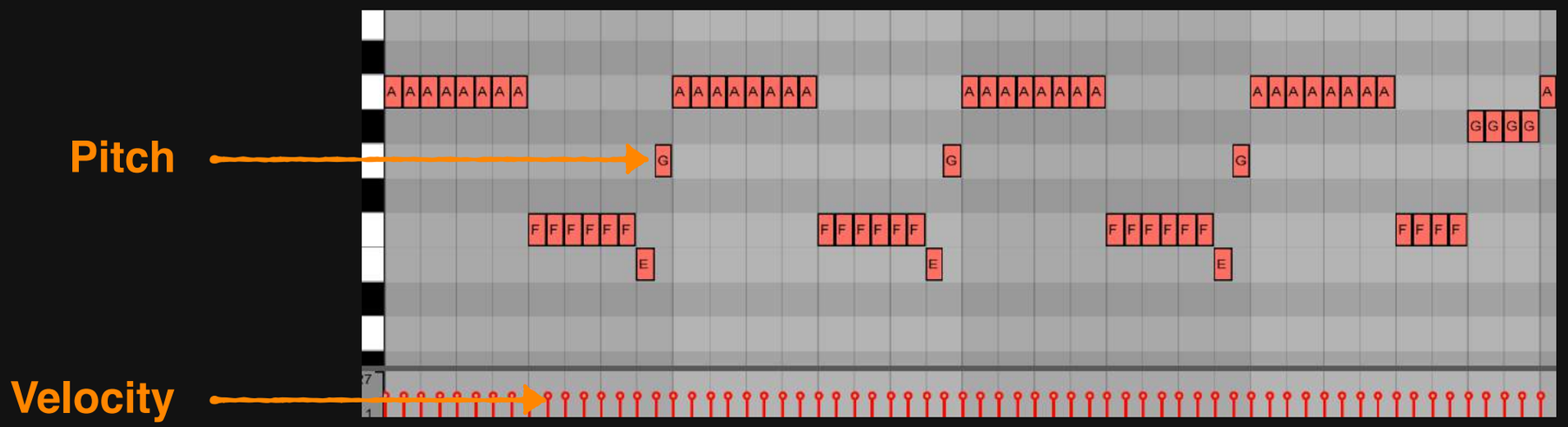
协.议

数字音乐设备之间彼此通信的协议

计算机可以理解的乐谱

音符 + 音量

MIDI Track



Interactive Electro-acoustic Music

MIDI

格式化

command：1001 = ON 1000 = OFF 1110 = Pitch bend

channel：16 channels 0011 = 4

pitch: 0 - 127

velocity: 0 - 127

MIDI

格式化

音高数值对照表

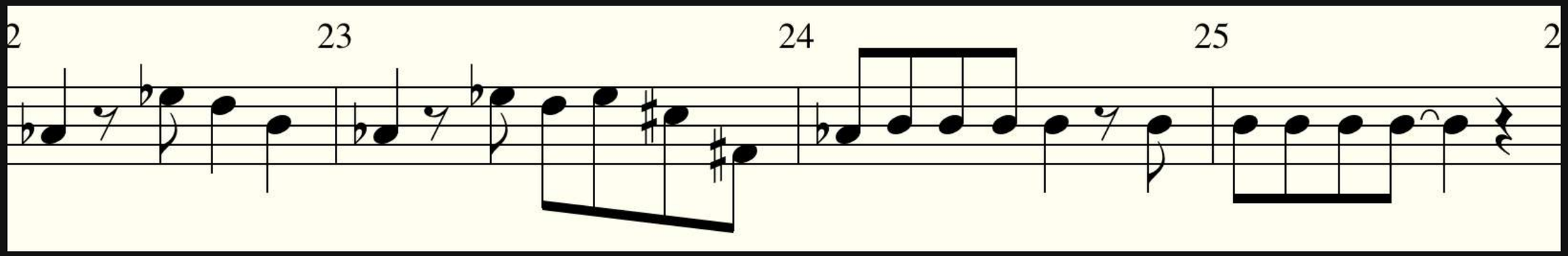
音高	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-2	00	01	02	03	04	05	06	07	08	09	10	11
-1	12	13	14	15	16	17	18	19	20	21	22	23
0	24	25	26	27	28	29	30	31	32	33	34	35
1	36	37	38	39	40	41	42	43	44	45	46	47
2	48	49	50	51	52	53	54	55	56	57	58	59
3	60	61	62	63	64	65	66	67	68	69	70	71
4	72	73	74	75	76	77	78	79	80	81	82	83
5	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107
7	108	109	110	111	112	113	114	115	116	117	118	119
8	120	121	122	123	124	125	126	127				

Interactive Electro-acoustic Music

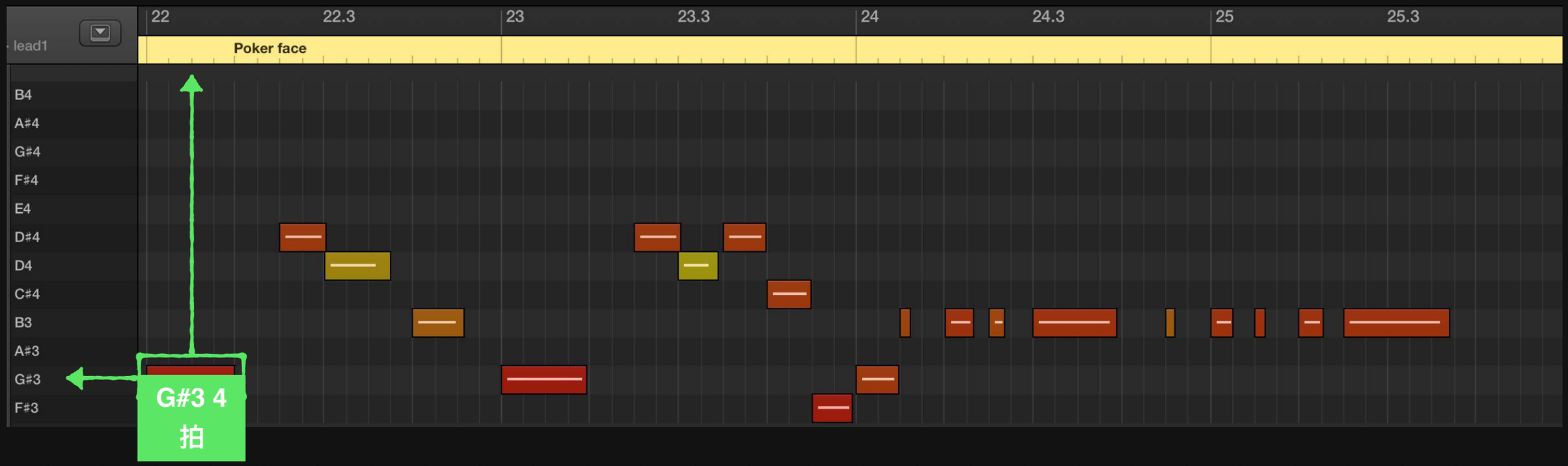
MIDI

翻译

乐谱



MIDI谱



Pitch	G#3	0	D#4	D4	0	B3	0	G#3	0	D#4	D4	D#4	C#4	F#3	G#3	B3	0	B3	0	B3	0	B3	0	B3	0	B3	0	B3	0	B3
Delay	4	2	2	3	1	2	2	4	2	2	2	2	2	2	2	0.5	1.5	1	1	1	1	4	2	0.5	1.5	1	1	0.5	1.5	5
MIDI	68		75	74		71		68		75	74	75	74	65	68	71		71		71		71		71		71		71		71

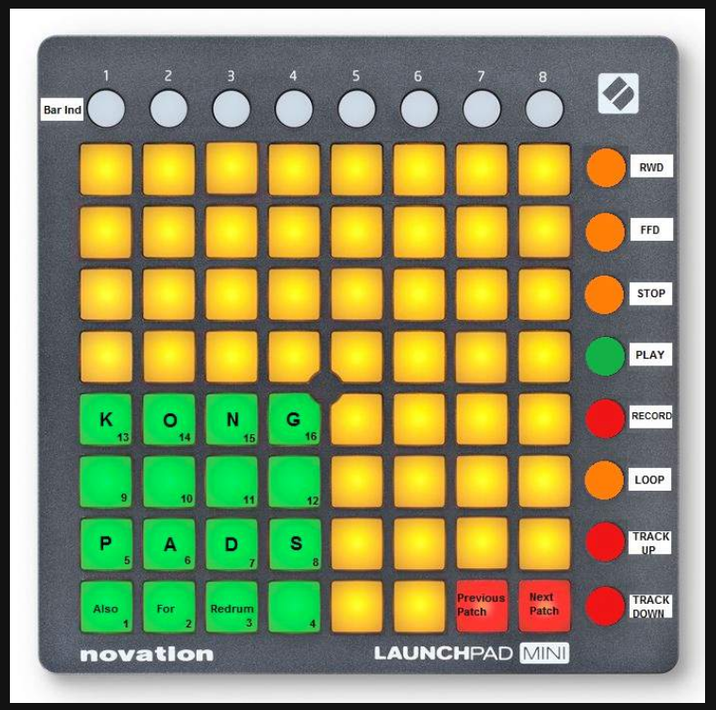
Interactive Electro-acoustic Music

MIDI

控制器

MIDI Keyboard

Launchpad



Synthesizer

MIDI

发.射

Send MIDI message with Arduino

1

定义 MIDI 命令

```
int noteON = 144;
int noteOFF = 128;
```

2

定义 Note & Velocity

```
int velocity = 100;
int note = 60;
```

3

定义 MIDImessage() 函数

```
void MIDImessage(int command, int MIDInote, int MIDIvelocity) {
  Serial.write(command);
  Serial.write(MIDInote);
  Serial.write(MIDIvelocity);
}
```

4

在 loop() 里调用 MIDImessage() 函数

```
MIDImessage(noteON, note, velocity);
MIDImessage(noteOFF, note, velocity);
```

MIDI

发.射

```
int noteON = 144;   // 144 = 10010000 in binary, note on command
int noteOFF = 128;   // 128 = 10000000 in binary, note off command
int velocity = 100;   // between 0 and 127

void setup() {
Serial.begin(9600);    //  set MIDI baud rate
}

void loop() {
  for (int note=48;note<72;note++) {
    MIDImessage(noteON, note, velocity);   // turn note on
    delay(250);   // hold note for 300ms
    MIDImessage(noteOFF, note, velocity);   // turn note off
    delay(250);   // wait 200ms until triggering next note
  }
}

void MIDImessage(int command, int MIDInote, int MIDIvelocity) {
  Serial.write(command);   // send note on or note off command
  Serial.write(MIDInote);   // send pitch data
  Serial.write(MIDIvelocity);   // send velocity data
}
```

Interactive Electro-acoustic Music

MIDI

发.射

Highlight

```
MIDImessage(noteON, note, velocity);
```

MIDI 信息的三要素：命令，音符，音量

```
delay(300);
```

音符持续的时间，使用变量控制以改变节奏

```
Serial.write(command);
```

将单个字节数据写入串口

MIDI

发.射

模拟传感器输入控制 note

```
int noteON = 144;   // note on command
int analogPin = A0;   // analog sensor connects to A0
int velocity = 100;

void setup() {
Serial.begin(9600);    //  set MIDI baud rate
}

void loop() {
  int val = analogRead(analogPin);   // read data from analog sensor
  int note = map(val, 0, 1023, 0, 127);    // scale data to fit the range of MIDI notes
  MIDImessage(noteON, note, velocity);   // turn note on
  delay(300);   // hold note for 300ms
  MIDImessage(noteON, note, 0);   // turn note off, note on with velocity 0
  delay(200);   // wait 200ms until triggering next note
}

void MIDImessage(int command, int MIDInote, int MIDIvelocity) {
  Serial.write(command);   // send note on or note off command
  Serial.write(MIDInote);   // send pitch data
  Serial.write(MIDIvelocity);   // send velocity data
}
```

MIDI

发.射

Highlight

```
int note = map(val, 0, 1023, 0, 127);
```

把模拟输入值范围映射到 MIDI Note 的范围 [0,127]

注意：不同模拟传感器范围可能不同

```
MIDImessage(noteON, note, 0);
```

Velocity = 0 的 noteON 等同于 noteOFF

MIDI

发.射

数字传感器触发 MIDI

```
int noteON = 144;   // note on command
int digitalPin = 2;   // digital sensor connects to D2
int velocity = 100;

void setup() {
pinMode(digitalPin, INPUT);
Serial.begin(9600);    //  set MIDI baud rate
}
void loop() {
  int val = digitalRead(digitalPin);   // read data from digital sensor
  if (val == HIGH) {       // when sensor is triggered on
    MIDImessage(noteON, note, velocity);   // turn note on
    delay(250);   // hold note for 300ms
    MIDImessage(noteON, note, 0);   // turn note off, note on with velocity 0
    delay(250);   // wait 200ms until triggering next note
  }
}
void MIDImessage(int command, int MIDInote, int MIDIvelocity) {
  Serial.write(command);   // send note on or note off command
  Serial.write(MIDInote);   // send pitch data
  Serial.write(MIDIvelocity);   // send velocity data
}
```

MIDI

数字触发后发送一段旋律

发.射

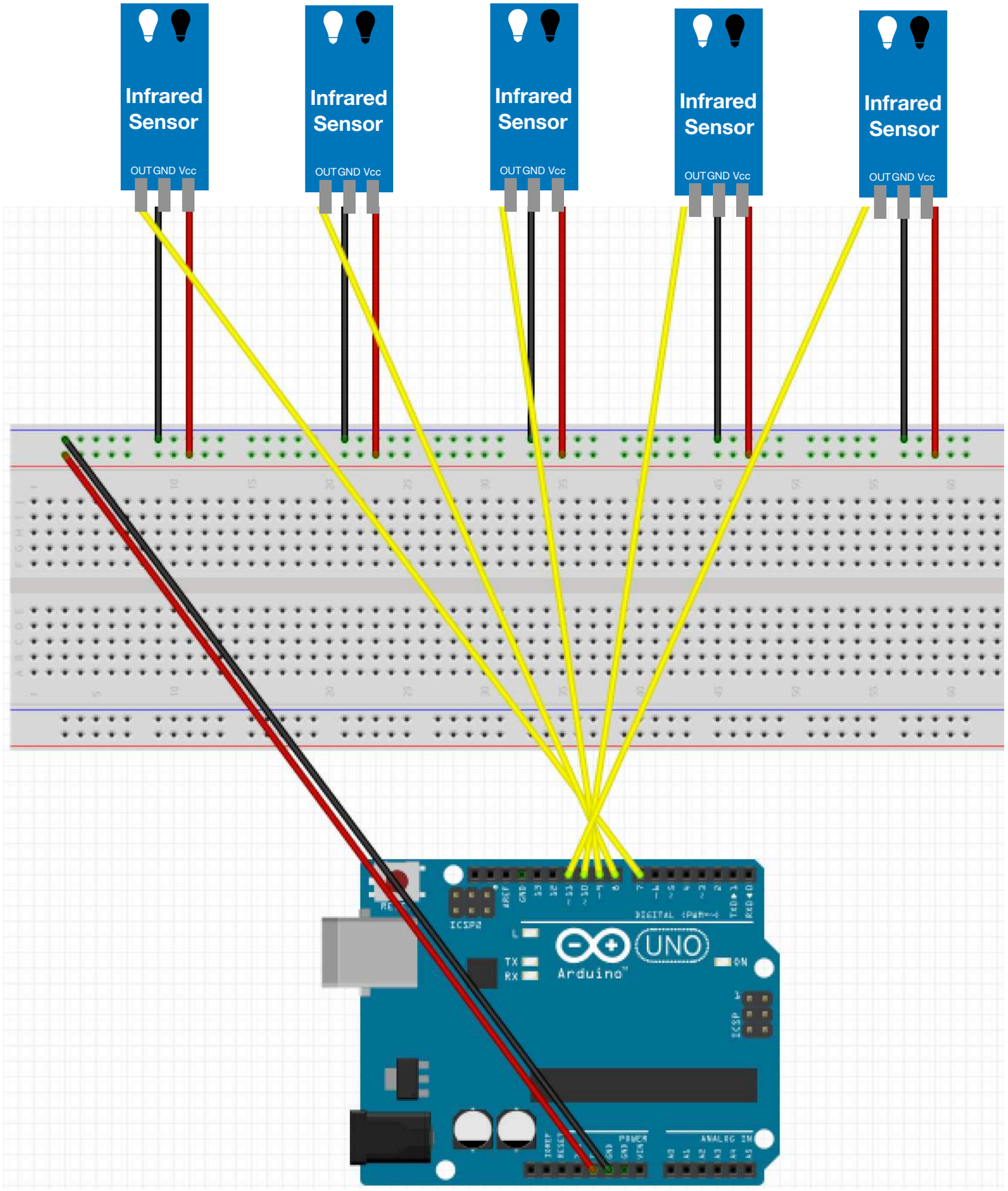
```
void loop() {
  int val = digitalRead(digitalPin);   // read data from digital sensor
  if (val == HIGH) {       // when sensor is triggered on
    MIDImessage(noteON, Dd4, velocity);//turn note on
    delay(250);//hold note for 250ms 2div
    MIDImessage(noteON, D4, velocity);
    delay(375);
    MIDImessage(noteON, 0, velocity);
    delay(125);
    MIDImessage(noteON, B3, velocity);
    delay(250);
    MIDImessage(noteON, 0, velocity);
    delay(250);
    MIDImessage(noteON, Gg3, velocity);
    delay(500);
    MIDImessage(noteON, 0, velocity);
    delay(250);
    MIDImessage(noteON, Dd4, velocity);
    delay(250);
    MIDImessage(noteON, D4, velocity);
    delay(250);
    MIDImessage(noteON, Dd4, velocity);
```

MIDI

使用模拟传感器数值实时控制音符音量（velocity）

发.射

```
void loop() {
   MIDImessage(noteON, Gg3, map(analogRead(A0), 240, 420, 20, 127));
   // map函数将模拟传感器数值的最小值到最大值的区间映射到音量的范围（0-127）内
   delay(500);//hold note for 300ms
   MIDImessage(noteON, 0, 0));//turn note off
   delay(250);//hold note for 300ms
}
void MIDImessage(int command, int MIDInote, int MIDIvelocity) {
  Serial.write(command);   // send note on or note off command
  Serial.write(MIDInote);   // send pitch data
  Serial.write(MIDIvelocity);   // send velocity data
}
```


Infrared Sensor
OUT GND Vcc
Infrared Sensor
OUT GND Vcc
Infrared Sensor
OUT GND Vcc
Infrared Sensor
OUT GND Vcc
Infrared Sensor
OUT GND Vcc
UNO
Arduino

红外

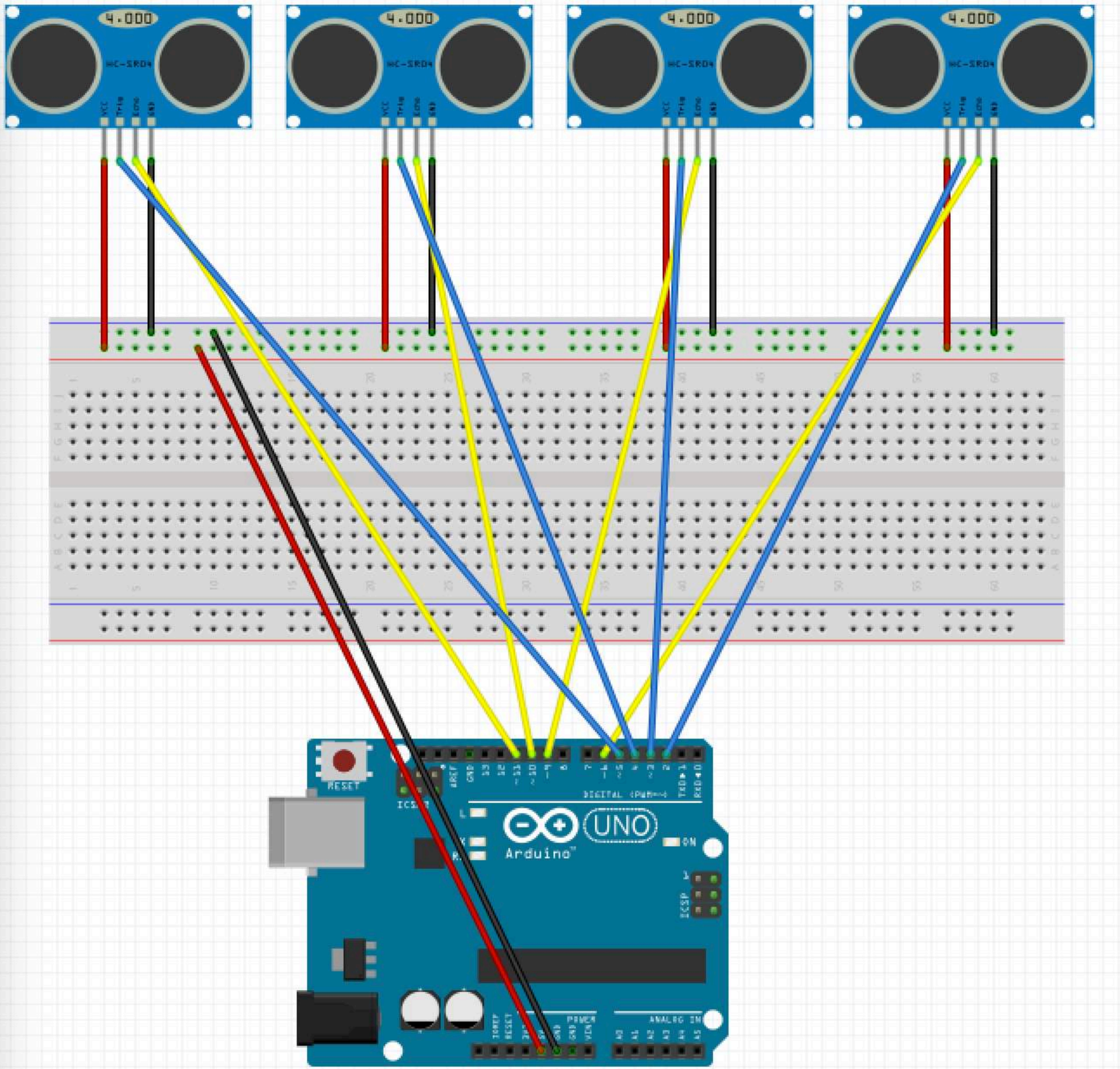
2 Arduino 编程

```
int infraredPin[]={7,8,9,10,11};
int noteON=144;
int noteOFF=128;

void setup() {
  for(int i=0;i<5;i++){
    pinMode(infraredPin[i],INPUT);
  }
  Serial.begin(9600);
}

void loop() {
  for(int i=0;i<5;i++){
    if(digitalRead(infraredPin[i])==0){
    MIDImessage(noteON,60,80);
    }
  }
}
```

超声波


4.000
HC-SR04
VCC
Trig
Echo
GND
RESET
UNO
Arduino
DIGITAL (PWM~)
POWER
ANALOG IN

Arduino 编程

```
void loop() {

  for (int i = 0; i < 4; i++) {
    digitalWrite(TrigPin[i], LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin[i], HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPin[i], LOW);
    distance[i]= pulseIn(EchoPin[i], HIGH) / 58.0;
  }

}
```

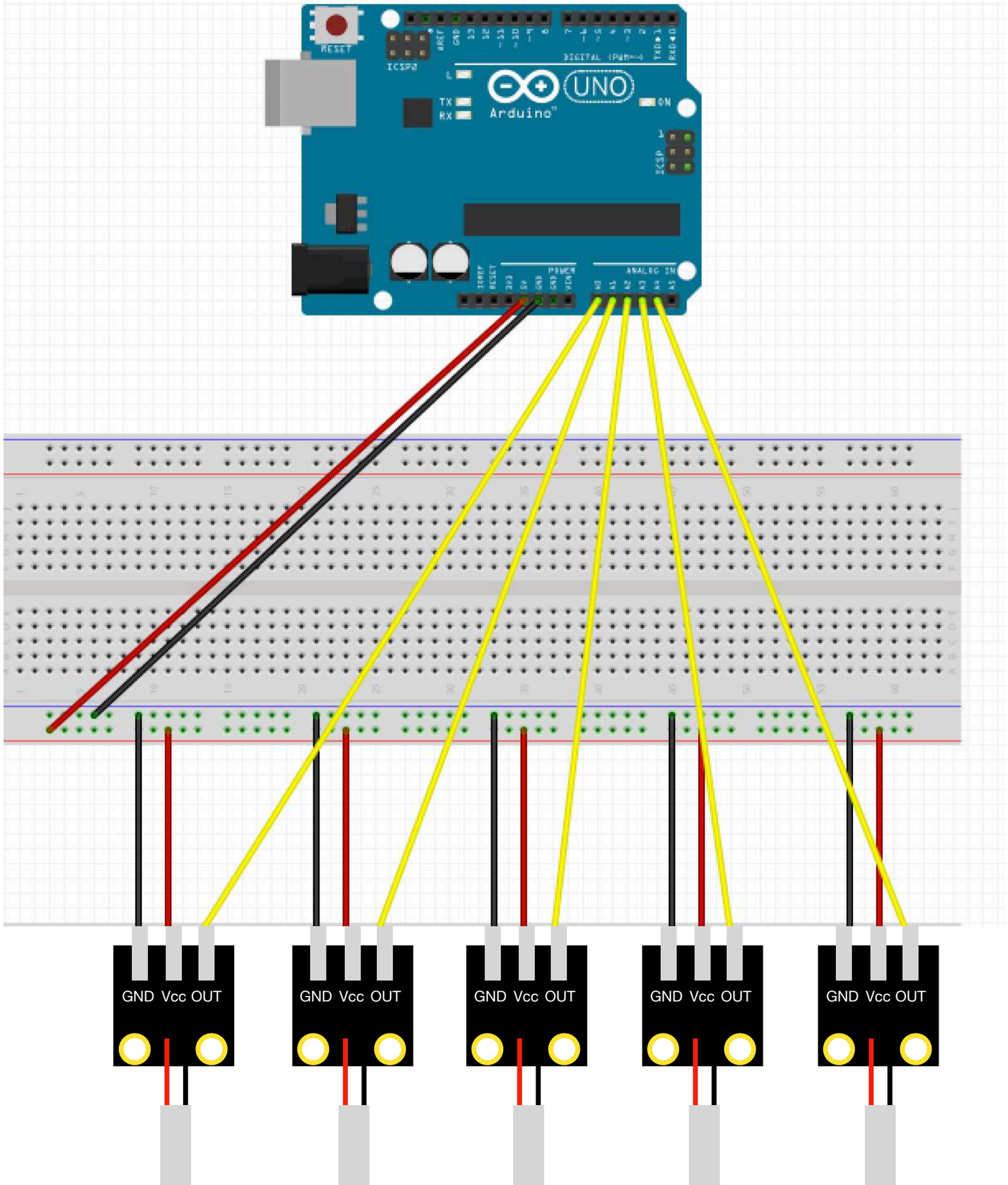
2 Arduino 编程

```
if (distance[3] > 3 && distance[3] < 40) {
    int velocity=map(distance[3],0,40,0,127);

    if (distance[0] > 3 && distance[0] < 40) {
      int note1=map(distance[0],0,40,36,47);
      MIDImessage(144,note1,velocity);
      delay(300);
      MIDImessage(128,note1,0);
    }
    if (distance[1] > 3 && distance[1] < 40) {
      int note2=map(distance[1],0,40,48,59);
      MIDImessage(145,note2,velocity);
      delay(300);
      MIDImessage(129,note2,0);
    }
    if (distance[2] > 3 && distance[2] < 40) {
      int note3=map(distance[2],0,40,60,71);
      MIDImessage(146,note3,velocity);
      delay(300);
      MIDImessage(130,note3,0);
    }

}
```

柔性压电

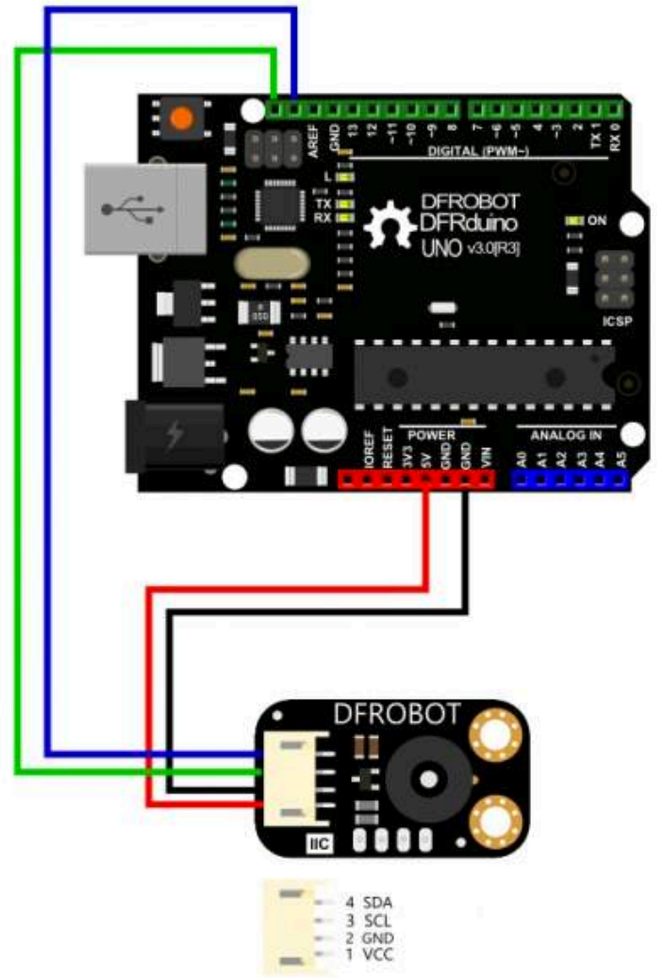

RESET
ICSP2
UNO
Arduino
TX
RX
ON
ICSP
POWER
ANALOG IN
GND Vcc OUT
GND Vcc OUT
GND Vcc OUT
GND Vcc OUT
GND Vcc OUT

Arduino 编程

```
for(int i=0;i<5;i++){
      n[i]=analogRead(drumPin[i]);
      velocity[i]=map(n[i],50,450,0,127);
    }

      //the first drum
      if(n[0]>=100 && n[0]<200){
        MIDImessage(144,24,velocity[0]);
        delay(300);
        MIDImessage(128,24,velocity[0]);
}
 else if(n[0]>=200 && n[0]<300){
 MIDImessage(144,48,velocity[0]);
 delay(300);
  MIDImessage(128,48,velocity[0]);
}
else{
  MIDImessage(144,72,velocity[0]);
  delay(300);
  MIDImessage(128,72,velocity[0]);
}
```

1. 硬件连接：将 SKT 传感器按下图的连线方式
   连接在UNO板扩展板上的模拟端口 A2



2 Arduino 编程

```
void loop() {
  Temp = MLX90614.GetObjectTemp_Celsius();
  Atemp = MLX90614.GetAmbientTemp_Celsius();
  note = map(Temp,30,31,24,60);
  velocity = map(Atemp,30,31,24,90);
  if(Temp>30){
  MIDImessage(144,note,velocity);
  delay(100);
  MIDImessage(128,note,velocity);
  }
}
```

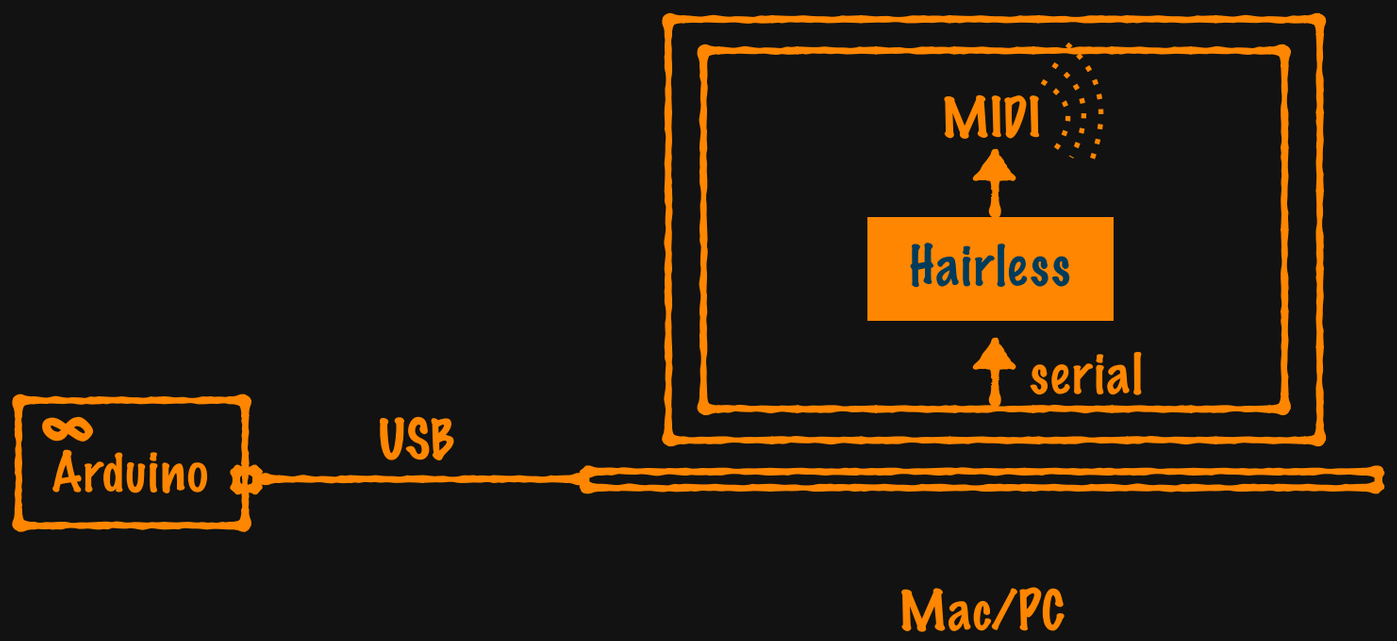
Interactive Electro-acoustic Music

MIDI

虚拟语气

虚拟 MIDI 桥

Serial —> MIDI



Download Hairless for Mac/PC

HOW TO DO

1 **修改比特率**

选择 preference 修改 Baud rate 为 9600

确保和 IDE 里串口的比特率相同，如果不相同 Hailess 会报错

2 **打开 Mac 自带的 IAC Driver**

Mac/Apps 打开 Audio MIDI Setup

Window 里 Show MIDI Studio

双击灰掉的 IAC Driver

勾选 device is online

PC 使用 LoopMIDI 软件 点“+”
添加一个 MIDI Port

3 **选择串口和 MIDI OUT**

Serial Port 选自己的 Arduino Board

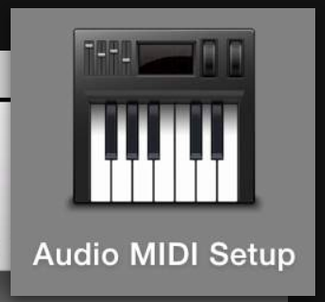
MIDI OUT 选 IAC Driver Bus1 (Mac)

LoopMIDI 1 (PC)

4 **运行 Hairless**

勾选 Serial<->MIDI bridge on

注意：upload 代码前先断开连接，否则 Arduino IDE 会报错



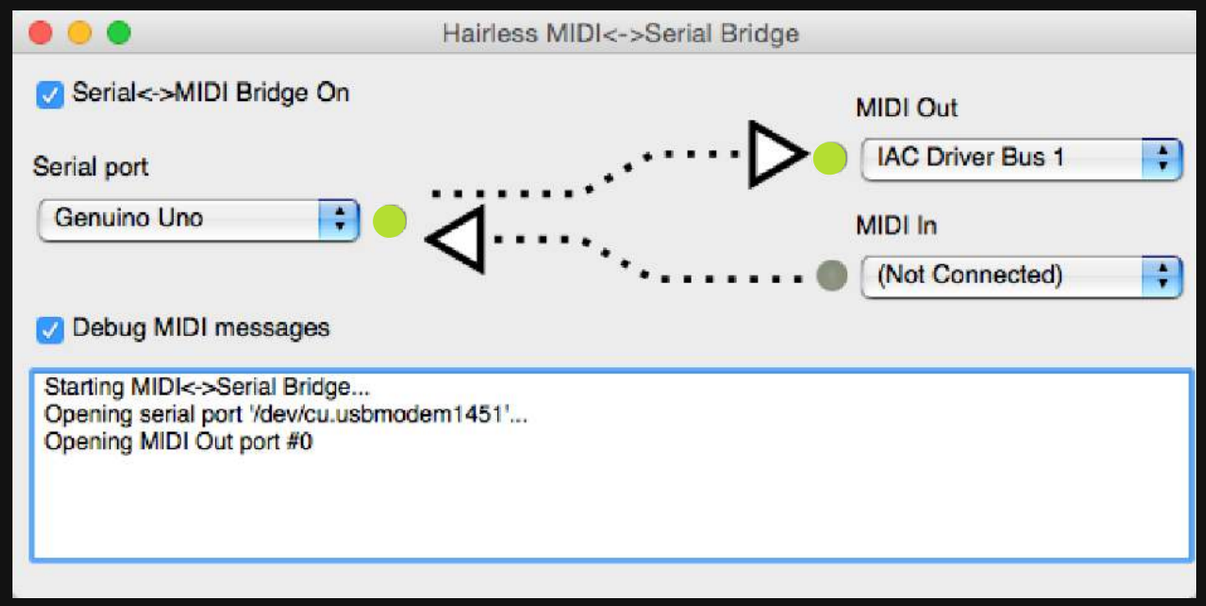
Interactive Electro-acoustic Music

MIDI

虚拟语气

修改比特率

运行时接收到 MIDI 信息时，信号灯会闪烁，Debug 会显示转换过的 command, note, velocity



Hairless Bridge 主界面