BMduino-Shield
MIDI 扩充板

BMV51T001
Arduino Library V1.0.1 说明

版本：V1.00　日期：2023-08-01

www.bestmodulescorp.com

目录

简介

BMV51T001 是倍创推出的自带波表合成的 MIDI 音色库扩充板，使用 UART 通信方式进行命令收发。本文档对 BMV51T001 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例演示了琴键和打击功能，以及 MIDI 歌曲的播放功能。

Arduino Library 函数

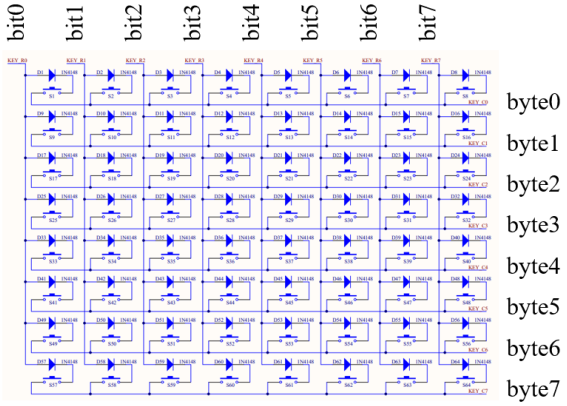
Arduino Lib 名称：BMV51T001		Lib 版本：V1.0.1
构造函数 & 初始化		
1	BMV51T001 ()	
	描述	构造函数
	参数	—
	返回值	—
	备注	—
2	void begin(void)	
	描述	扩充板初始化
	参数	void
	返回值	void
	备注	—
功能函数		
3	bool setVolume(uint8_t volume)	
	描述	设置音量
	参数	volume ： 音量，范围 0~15，0 是最小音量 (静音)，默认是音量 5
	返回值	执行情况： true：成功 false：失败
	备注	—
4	bool setHitSensitivity(uint8_t value)	
	描述	设置打击灵敏度
	参数	value：打击灵敏度参数，范围 0~18 触发电压 = 3.3 × ((100+50×value)/4096)，单位：V
	返回值	执行情况： true：成功 false：失败
	备注	打击灵敏度设置请见使用手册 → 传感接口介绍。打击灵敏度参数默认 0。
5	bool setHitTimeInterval(uint8_t value)	
	描述	设置打击检测的间隔时间
	参数	value：间隔时间参数 ， 范围 0~33 间隔时间 = 35 + 5 × 间隔时间参数，单位：ms
	返回值	执行情况： true：成功 false：失败
	备注	打击灵敏度设置请见使用手册 → 传感接口介绍。间隔时间参数默认 3。

6	void setHitStrengthLayer(uint8_t layers)	
	描述	设定打击力度等级
	参数	layers：力度等级，范围 1~128
	返回值	void
	备注	—
7	bool setKeyboardOut(uint8_t status)	
	描述	设置 8×8-key 键盘功能状态
	参数	status：键盘功能状态 1 (BMV51T001_KEYBOARD_ENABLE)：使能 0 (BMV51T001_KEYBOARD_DISABLE)：除能
	返回值	执行情况： true：成功 false：失败
	备注	—
8	bool setHitOut (uint8_t status)	
	描述	设置打击功能状态
	参数	status：打击功能状态 1 (BMV51T001_HIT_ENABLE)：使能 0 (BMV51T001_HIT_DISABLE)：除能
	返回值	执行情况： true：成功 false：失败
	备注	—
9	void scanKeyboard(void)	
	描述	琴键功能扫描
	参数	void
	返回值	void
	备注	在 loop 里面调用
10	void scanHit(void)	
	描述	打击功能扫描
	参数	void
	返回值	void
	备注	在 loop 里面调用
11	bool isKeyboard(void)	
	描述	琴键是否有动作
	参数	void
	返回值	动作情况 true：有动作 false：没动作
	备注	—

12	bool isHit(void)	
	描述	打击是否有动作
	参数	void
	返回值	动作情况 true：有动作 false：没动作
	备注	—
13	void readKeyboardData(uint8_t keyBuf[])	
	描述	读取 8×8 矩阵琴键数据
	参数	keyBuf[]：8×8 琴键状态，共 8byte，每个 bit 表示一个按键 bit=1：按下，bit=0：松开
	返回值	—
	备注	8×8 矩阵按键用 8 个 byte 存储，每个 bit 表示一个按键 (1)。
14	uint16_t readHitADCData(void)	
	描述	读取打击检测 ADC 数据
	参数	void
	返回值	16-bit 打击的 ADC 数据
	备注	—
15	uint8_t getHitStrengthLayer(uint16_t HhitADCData)	
	描述	转化当前的 ADC 数值为打击力度等级
	参数	HitADCData：16-bit ADC 数据
	返回值	力度等级：1~128
	备注	—
16	bool reset(void)	
	描述	复位扩充板
	参数	void
	返回值	复位情况 true：成功 false：失败
	备注	—
17	uint8_t getFWVer(void)	
	描述	获取版本号
	参数	void
	返回值	版本号
	备注	目前版本为 0x01 版
MIDI 通信协议函数		
18	void setNoteOn (uint8_t noteNumber, uint8_t velocity, uint8_t channel)	
	描述	发送音符开
	参数	noteNumber：音符，范围 0~127 velocity：力度，范围 0~127 channel：通道，范围 1~16 (打击乐器音色固定在通道 10)
	返回值	void
	备注	当使用通道 10 时为打击乐，noteNumber 范围 24~84，打击乐种类见附录一

19	void setNoteOff (uint8_t noteNumber, uint8_t velocity, uint8_t channel)	
	描述	发送音符关
	参数	noteNumber：音符，范围 0~127 velocity：音符力度，范围 0~127 channel：通道，范围 1~16 (打击乐器音色固定在通道 10)
	返回值	void
	备注	当使用通道 10 时为打击乐，noteNumber 范围 24~84，打击乐种类见附录一
20	void setTone (uint8_t toneNumber, uint8_t channel)	
	描述	设置音色
	参数	toneNumbe：音色号，范围 0~127 channel：通道，范围 1~16
	返回值	void
	备注	音色号见附录一中的音色表
21	void setChannelVolume(uint8_t channelVolume, uint8_t channel)	
	描述	设置通道音量
	参数	channelVolume ：通道音量值，范围 0~127 (0：最小；127：最大) channel：通道，范围 1~16
	返回值	void
	备注	区别于板载按键上的音量调节，板载按键上的音量调节是调节整体音量，这里仅仅是通道音量
22	void setPitchBend(int16_t pitchValue, uint8_t channel)	
	描述	设置弯音音效
	参数	pitchValue：弯音值，范围 -8192~8191 (0：没有弯音效果，-8192：最大向下弯曲，8191：最大向上弯曲) channel：通道，范围 1~16
	返回值	void
	备注	—

(1) 矩阵按键，数据与按键的对应关系

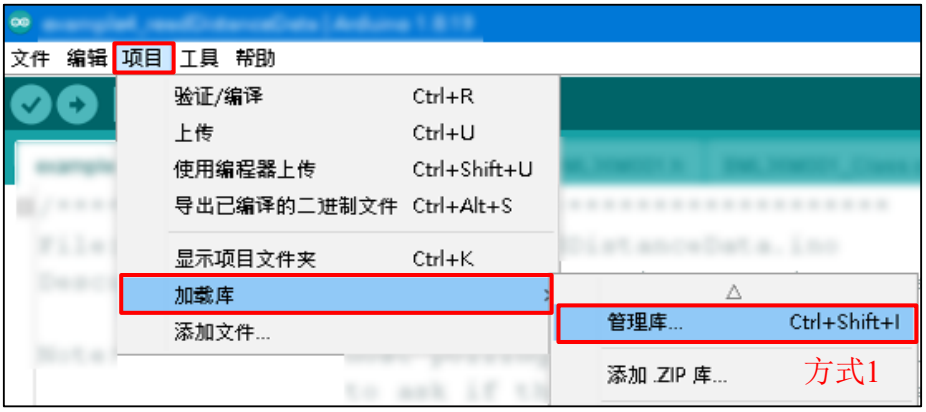


Arduino Lib 下载及安装

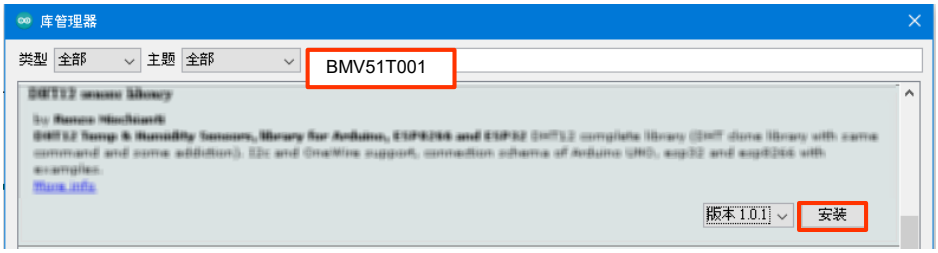
BMV51T001 Library：可参考下面两种方法安装 BMV51T001 的 Arduino Library

方式 1：搜索安装

搜索安装：Arduino IDE → 项目 → 载入库 → 管理库 → 搜索 BMV51T001 → 安装



搜索安装流程 **1**

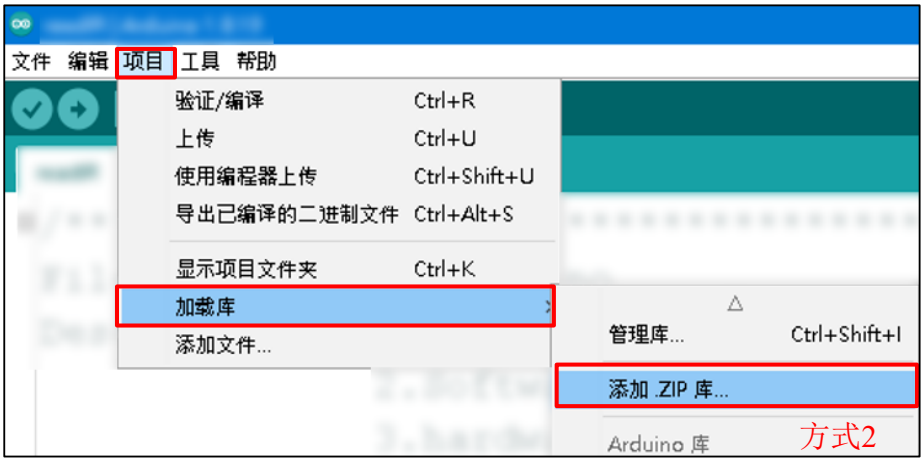


搜索安装流程 **2**

方式 2：添加 .ZIP 库，需提前下载 .ZIP 库

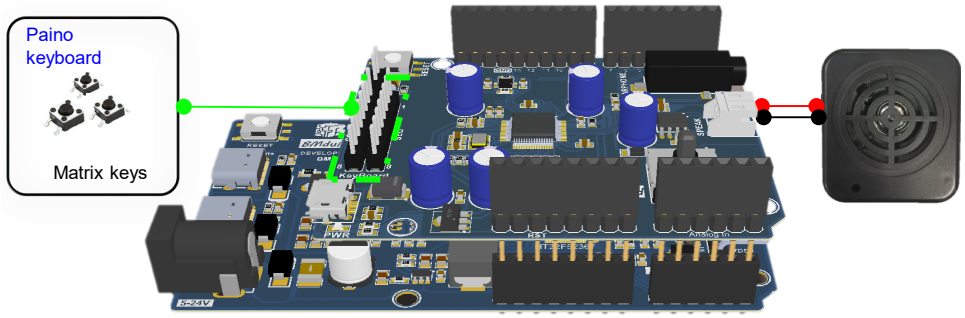
下载方法：打开倍创官方网站 (https://www.bestmodulescorp.com/bmv51t001.html#tab-product2) 文件目录下的 Arduino 范例程序 (BMV51T001 Library)。

添加 .ZIP 库：Arduino IDE → 项目 → 载入库 → 添加 .ZIP 库 ....



Arduino 范例

范例 1：PianoPerformance



實物連接示意圖

范例实现功能：通过 MIDI Shield 的琴键接口外接 64keys，然后触发按键发声。

1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BMV51T001) → 选择范例 (PianoPerformance)
2. 示例说明：
   a. 创建对象 & 声明功能函数 & 初始化扩充板

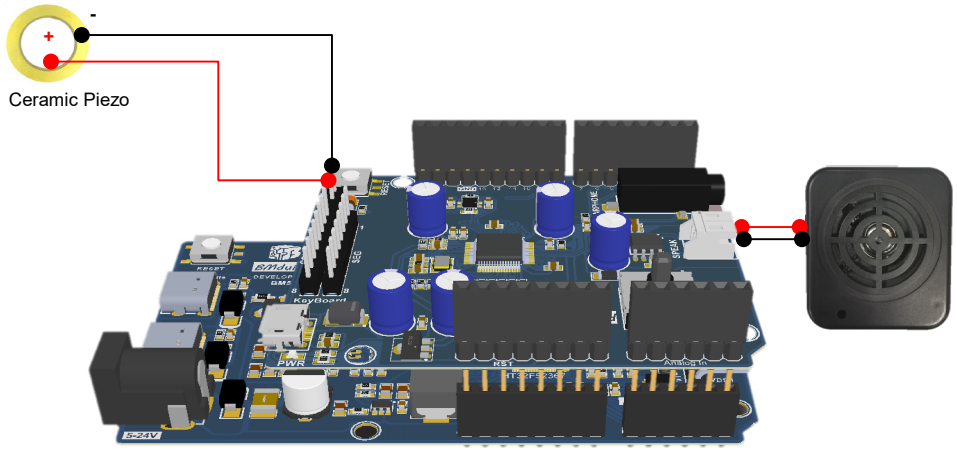
```
#include “BMV51T001.h”
BMV51T001 shieldMIDI;           // 创建对象
uint8_t keyBuf[8];  // 存放琴键数据，大小必须为 8，因为每次接收是 8byte 为单位
uint8_t lastKeyBuf[8];          // 存放上一次的琴键数据
uint8_t temp,i,j;
uint8_t noteNumber ;            // 音高
uint8_t velocity = 0x50;        // 力度，钢琴没做力度检测，默认 0x50
uint8_t channel = 0x01;         // 播放默认在 channel 1
uint8_t keyOutEnableFlag = 0; // 设置琴键输出标志

void setup()
{
  shieldMIDI.begin();           // 初始化扩充板
  if(shieldMIDI.setKeyboardOut(BMV51T001_KEYBOARD_ENABLE)) // 使能琴键
                                                           // 功能
  {
    keyOutEnableFlag = 1;
  }
}
```

b. 在 loop 中执行 PianoPerformance 功能函数

```
void loop()
{
  if(keyOutEnableFlag)
  {
    shieldMIDI.scanKeyboard();          //UART 接收按键扫描
    if(shieldMIDI.isKeyboard()!=0)      // 接收完成
    {
      shieldMIDI.readKeyboardData (keyBuf);// 读取按键状态
      for (i = 0; i < 8; i++)           //8 组 key 全部判断一遍
      {
        temp = keyBuf[i]^lastKeyBuf[i];// 前后两次按键有变化
        if (temp!=0)  // 如果有按键按下
        {
          for (j = 0; j < 8; j++)       // 每组 key 里面有 8 个 key
          {
            if ((temp &(1 << j))!=0)   // 该位置有按键变化
            {
              noteNumber = 36 +i*8+j;  // 由于它是 64 键，从 noteNumber
                                       // 36 大字一组的 do 开始
              if ((keyBuf[i]&(1 << j))!=0)  // 按键按下
              {
                // 调用 note on 播放音符
                shieldMIDI.setNoteOn (noteNumber, velocity, channel);
              }
              else // 按键松开
              {
                // 调用 note off 停止音符
                shieldMIDI.setNoteOff (noteNumber, velocity, channel);
              }
            }
          }
        lastKeyBuf[i]= keyBuf [i];
        }
      }
    }
  }
}
```

范例 2：HitPerformance



实物连接示意图

范例实现功能：读取压电陶瓷片的 ADC 数值，转成电子鼓发声力度。

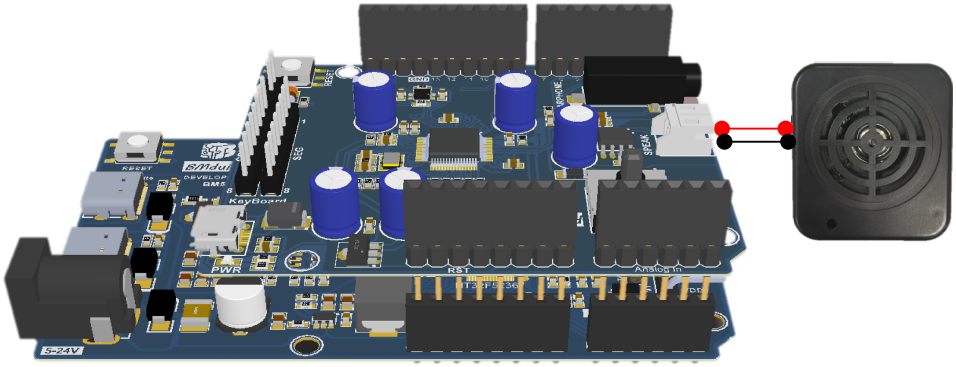
1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BMV51T001) → 选择范例 (HitPerformance)
2. 示例说明：
   a. 创建对象 & 声明功能函数 & 初始化扩充板

```
#include “BMV51T001.h”
BMV51T001 shieldMIDI;           // 创建对象
uint16_t hitADCData = 0;        // 存放读取到的打击数据
uint8_t velocity = 0;           // 力度
uint8_t noteNumber = 0x40;      // 定义为鼓声
uint8_t channel = 10;           // 鼓声固定在通道 10
uint8_t hitOutEnableFlag = 0; // 设置打击输出标志
#define BMV51T001_NO_HIT 0
void setup()
{
  shieldMIDI.begin();           // 初始化扩充板
  shieldMIDI.setHitStrengthLayer(128);// 设置力度等级，将总的力度分为
                                      // 128 级
  if(shieldMIDI.setHitOut(BMV51T001_HIT_ENABLE))// 使能打击检测功能
  {
    hitOutEnableFlag = 1;       // 使能打击功能
  }
}
```

b. 在 loop 中执行 HitPerformance 功能函数

```
void loop()
{
  if(hitOutEnableFlag)    // 打击功能已使能
  {
    shieldMIDI.scanHit();//UART 接收打击设备
    if(shieldMIDI.isHit()!= BMV51T001_NO_HIT)    // 接收完成
    {
       hitADCData =shieldMIDI.readHitADCData ();// 读取打击检测数据
       velocity = shieldMIDI. getHitStrengthLayer(hitADCData);// 转化
                                                    // 为力度等级
       shieldMIDI.setNoteOn (noteNumber, velocity, channel);// 发送音
        // 符开消息，通道 10 为鼓音色专用通道，如果发送通道 10 则表示这是鼓音色
    }
  }
}
```

范例 3：PlayMidi



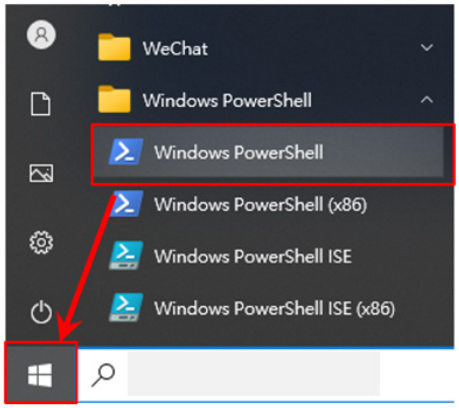
实物连接示意图

范例实现功能：播放 .mid 歌曲。

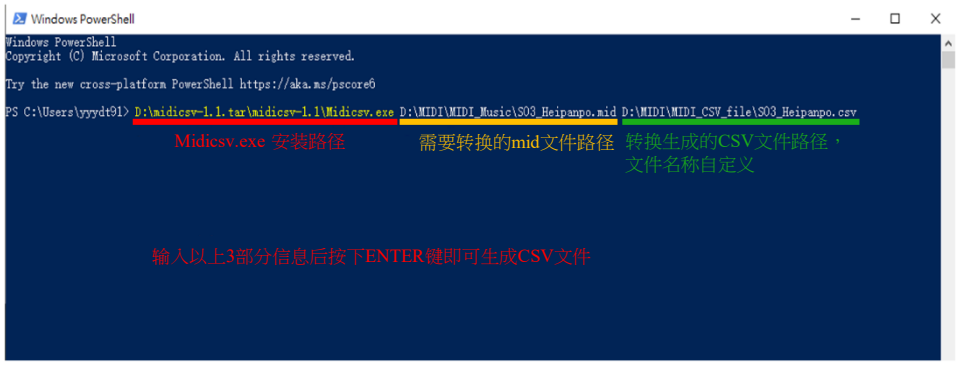
1. 提取 .mid 文件数据，供单片机使用

步骤 1：请下载 Midicsv.exe 软件。

步骤 2：打开计算机自带的 Windows PowerShell (支持 WIN11\WIN10\WIN7)，将 MIDI 文件转化成 CSV 表格文件。

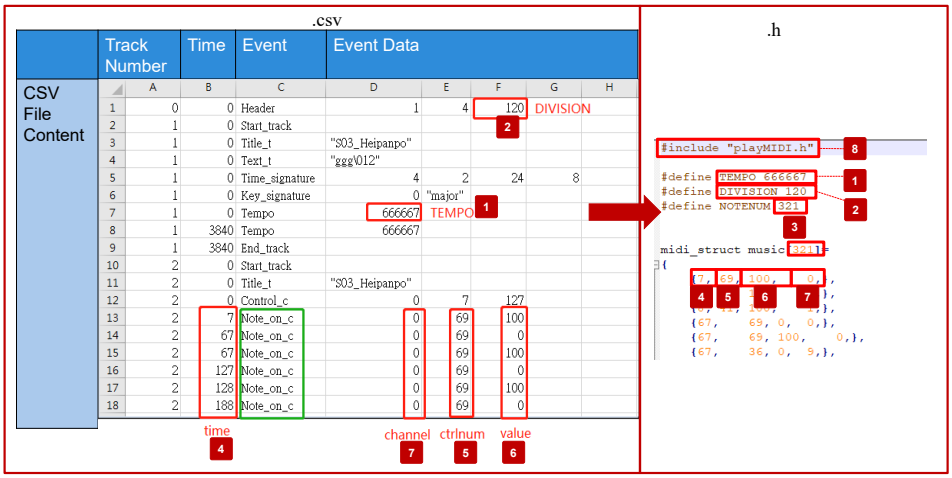


如下图 所示，在 PowerShell 中输入三个路径，路径之间用空格隔开① Midicsv.exe 路径② .mid 文件路径③ .csv 文件存储路径。输入完后按下 enter 键即可完成转换。



注意：上图中的路径需要根据用户实际路径填入

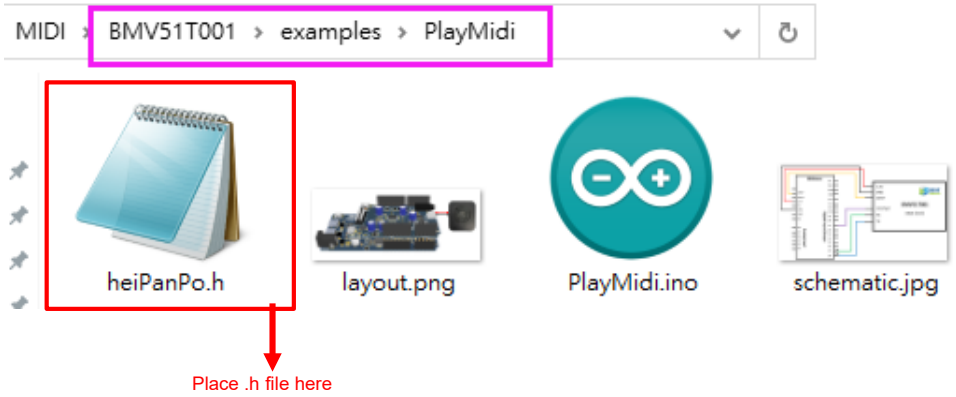
步骤 3：打开转化后的 .csv 文件，建立 .h 文件。将 .csv 文件的内容按下图建立数据，填入 .h 文件中，左框与右框中的内容按数字编号一一对应



上图中左边 .csv 制作成右边 .h 文件，其中各项内容为：

序号	名称	描述
1	TEMPO	一拍的时间 (μs 为单位)
2	DIVISION	一拍的 tick 数 (tick 是 midi 中计算时间长短的最小单位)
3	NOTENUM	音符数量，这个需要使用者自己根据 csv 里面 note_on_c 的行数填入
4	time	当前的音乐进度，以 tick 为单位的时间
5	ctrlnum	音符编号 (音符音高)
6	value	音符力度 (弹奏力度)
7	channel	通道 (音轨)
8	include	使用时必须 include “playMIDI.h” 头文件

步骤 4：.h 文件制作完毕后放到范例程序 .ino 文件的相同路径下，如下图



步骤 5：在 .ino 中添加上述制作的 .h 文件，单片机即可使用 midi 歌曲信息

2. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BMV51T001) → 选择范例 (PlayMidi)

3. 示例说明：

a. 创建对象 & 初始化扩充板

```
#include “BMV51T001.h”
#include “playMIDI.h”
#include “heiPanPo.h”
/* 工作原理：按照顺序播放音符列表，组成一首 MIDI 曲子，每个音符都有固定的时间，音色，
音高，力度，通道在列表里”heiPanPo.h”*/
playMIDI myMIDIPlayer; // 创建对象
void setup() {
  myMIDIPlayer.begin();// 初始化扩充板
  myMIDIPlayer.setChannelVolume(127, 0 + 1); //设置通道 1 音量，127 最大声。
  myMIDIPlayer.setTone(35, 1 + 1);  // 设置通道 2 播放的音色，35 是音色号
  myMIDIPlayer.setChannelVolume(127, 1 + 1); //设置通道 2 音量，127 最大声。
  myMIDIPlayer.setChannelVolume(127, 9 + 1); // 设置通道 10 音量，127 最大
  // 声。开始播放 .h 文档中数组名为 music 的 midi 歌曲
  myMIDIPlayer.beginPlayMIDI(music,TEMPO,DIVISION,NOTENUM);
}
```

b. 在 loop 中播放 Midi 曲

```
void loop() {
  myMIDIPlayer.loopPlayMIDI();// 扫描执行 MIDI 歌曲播放功能
  if(myMIDIPlayer.isPlaying()==MIDI_PLAYER_NO_BUSY)// 判断是否播放中
  {
    myMIDIPlayer.beginPlayMIDI(music,TEMPO,DIVISION,NOTENUM); // 重复
                                                              // 播放
  }
}
```

附录一：标准 MIDI 音色表

- 音色表

编号	名称	编号	名称
0	Acoustic Grand Piano	64	Soprano Sax
1	Bright Acoustic Piano	65	Alto Sax
2	Electric Grand Piano	66	Tenor Sax
3	Honky-tonk Piano	67	Baritone Sax
4	Electric Piano 1	68	Oboe
5	Electric Piano 2	69	English Horn
6	Harpsichord	70	Bassoon
7	Clavichord	71	Clarinet
8	Celesta	72	Piccolo
9	Glockenspiel	73	Flute
10	Music box	74	Recorder
11	Vibraphone	75	Pan Flute
12	Marimba	76	Blown Bottle
13	Xylophone	77	Shakuhachi
14	Tubular Bell	78	Whistle
15	Dulcimer	79	Ocarina
16	Hammond Organ	80	Lead 1(square)
17	Percussive Organ	81	Lead 2(sawtooth)
18	Rock Organ	82	Lead 3(calliope)
19	Church organ	83	Lead 4(chiff)
20	Reed organ	84	Lead 5(charang)
21	Accordion	85	Lead 6(voice)
22	Harmonica	86	Lead 7(fifths)
23	Tango Accordion	87	Lead 8(bass + lead)
24	Acoustic Guitar(nylon)	88	Pad 1(new age)
25	Acoustic Guitar(steel)	89	Pad 2(warm)
26	Electric Guitar(jazz)	90	Pad 3(polysynth)
27	Electric Guitar(clean)	91	Pad 4(choir)
28	Electric Guitar(muted)	92	Pad 5(bowed)
29	Overdriven Guitar	93	Pad 6(metallic)
30	Distortion Guitar	94	Pad 7(halo)
31	Guitar harmonics	95	Pad 8(sweep)
32	Acoustic Bass	96	FX 1(rain)
33	Electric Bass(finger)	97	FX 2(soundtrack)
34	Electric Bass(pick)	98	FX 3(crystal)
35	Fretless Bass	99	FX 4(atmosphere)
36	Slap Bass 1	100	FX 5(brightness)
37	Slap Bass 2	101	FX 6(goblins)

编号	名称	编号	名称
38	Synth Bass 1	102	FX 7(echoes)
39	Synth Bass 2	103	FX 8(sci-fi)
40	Violin	104	Sitar
41	Viola	105	Banjo
42	Cello	106	Shamisen
43	Contrabass	107	Koto
44	Tremolo Strings	108	Kalimba
45	Pizzicato Strings	109	Bagpipe
46	Orchestral Harp	110	Fiddle
47	Timpani	111	Shanai
48	String Ensemble 1	112	Tinkle Bell
49	String Ensemble 2	113	Agogo
50	Synth Strings 1	114	Steel Drums
51	Synth Strings 2	115	Woodblock
52	Voice Aahs	116	Taiko Drum
53	Voice Oohs	117	Melodic Tom
54	Synth Voice	118	Synth Drum
55	Orchestra Hit	119	Reverse Cymbal
56	Trumpet	120	Guitar Fret Noise
57	Trombone	121	Breath Noise
58	Tuba	122	Seashore
59	Muted Trumpet	123	Bird Tweet
60	French horn	124	Telephone Ring
61	Brass Section	125	Helicopter
62	Synth Brass 1	126	Applause
63	Synth Brass 2	127	Gunshot

● 打击乐

编号	名称	编号	名称
24	Seq Click H	55	Splash Cymbal
25	Brush Tap	56	Cowbell
26	Brush Swirl	57	Crash Cymbal 2
27	Brush Slap	58	Vibraslap
28	Brush Tap Swirl	59	Ride Cymbal 2
29	Snare Roll	60	Hi Bongo
30	Castanet	61	Low Bongo
31	Snare H Soft	62	Mute Hi Conga
32	Sticks	63	Open Hi Conga
33	Bass Drum Soft	64	Low Conga
34	Open Rim Shot	65	High Timbale
35	Acoustic Bass Drum	66	Low Timbale
36	Bass Drum 1	67	High Agogo
37	Side Stick	68	Low Agogo
38	Acoustic Snare	69	Cabasa
39	Hand Clap	70	Maracas
40	Electric Snare	71	Short Whistle
41	Low Floor Tom	72	Long Whistle
42	Closed Hi-Hat	73	Short Guiro
43	High Floor Tom	74	Long Guiro
44	Pedal Hi-Hat	75	Claves
45	Low Tom	76	Hi Wood Block
46	Open Hi-Hat	77	Low Wood Block
47	Low-Mid Tom	78	Mute Cuica
48	Hi-Mid Tom	79	Open Cuica
49	Crash Cymbal 1	80	Mute Triangle
50	High Tom	81	Open Triangle
51	Ride Cymbal 1	82	Shaker
52	Chinese Cymbal	83	Jingle Bell
53	Ride Bell	84	Bell Tree
54	Tambourine		