

BMduino-Shield
语音播放扩充板

BMV31T001
Arduino Library V1.0.1 说明

版本: V1.10 日期: 2024-01-24

www.bestmodulescorp.com

目录

简介	3
Arduino Library 函数	3
Arduino Lib 下载及安装	6
Arduino 范例	7
范例: voiceUpdateAndPlayback	7

简介

BMV31T001 是倍创推出的一款语音播放扩充板，采用单线通信进行播放控制。本文档对 BMV31T001 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例演示了在线音源更新，语音播放操作。

Arduino Library 函数

Arduino Lib 名称：BMV31T001		Lib 版本：V1.0.1	
构造函数 & 初始化			
1	BMV31T001()		
	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	void setVolume(uint8_t volume)		
	描述	设置音量	
	参数	volume: 音量参数，范围 0~11	
	返回值	void	
	备注	0 是最小音量 (静音)；总共有 12 级音量调节；内部默认音量是 8	
4	void playVoice(uint8_t num, uint8_t loop=0)		
	描述	播放语音	
	参数	num: 烧录音源时由烧录 PC 工具产生的语音序号 loop: 循环播放控制 0: 不循环 (默认) 1: 循环	
	返回值	void	
	备注	—	
5	void playSentence(uint8_t num, uint8_t loop=0)		
	描述	播放语句	
	参数	num: 烧录音源时由 PC tool 加载的语句顺序 loop: 循环播放控制 0: 不循环 (默认) 1: 循环	
	返回值	void	
	备注	—	

6	void playStop(void)	
	描述	停止播放当前语音 / 语句
	参数	void
	返回值	void
	备注	—
7	void playPause(void)	
	描述	暂停播放当前语音 / 语句
	参数	void
	返回值	void
	备注	—
8	void playContinue(void)	
	描述	如果有语音 / 句子处于暂停中，则继续播放，没有则无效
	参数	void
	返回值	void
	备注	—
9	void playRepeat(void)	
	描述	循环播放当前语音 / 语句
	参数	void
	返回值	void
	备注	—
10	bool isPlaying(void)	
	描述	检测是否在播放状态
	参数	—
	返回值	播放状态： true: 播放中 false: 不在播放中
	备注	—
11	void scanKey(void)	
	描述	扫描按键，在 loop 里面执行
	参数	void
	返回值	void
	备注	—
12	bool isKeyAction(void)	
	描述	判断按键状态有没有变化
	参数	—
	返回值	按键状态 true: 有变化 false: 没变化
	备注	按键状态有变化：按下 – 松开，松开 – 按下

13	uint8_t readKeyValue(void)	
	描述	读取按键状态
	参数	void
	返回值	按键的状态： 0x00: 没按键按下 0x01 (BMV31T001_KEY_MIDDLE): 中键有按下 0x02 (BMV31T001_KEY_UP): 上键有按下 0x04 (BMV31T001_KEY_DOWN): 下键有按下 0x08 (BMV31T001_KEY_LEFT): 左键有按下 0x10 (BMV31T001_KEY_RIGHT): 右键按下
	备注	—
14	void setPower(uint8_t status)	
	描述	设置扩充板的语音播放电路电源状态
	参数	status: 开关状态 0 (BMV31T001_POWER_DISABLE): 关闭 1 (BMV31T001_POWER_ENABLE): 打开
	返回值	void
	备注	—
15	void setLED(uint8_t status)	
	描述	设置状态指示灯 (绿色 LED) 状态
	参数	status: LED 状态 0: 暗 1: 亮
	返回值	void
	备注	—
在线音源更新函数		
16	void initAudioUpdate(unsigned long baudrate=256000)	
	描述	初始化在线更新音源
	参数	baudrate: 波特率, 默认 256000
	返回值	void
	备注	如果用户搭配 Arduino UNO 是通过 Serial 更新音源; 如果搭配 BMduino 是通过 NATIVE USB 虚拟串口更新音源。详情请参考使用手册 Arduino 兼容性章节
17	bool isUpdateBegin(void)	
	描述	是否有音源需要更新
	参数	void
	返回值	是否有音源需要更新: true: 有音源需要更新 false: 无音源需要更新
	备注	BMduino Voice Widge 上点击  下载后, 此函数可判断是否开发板有接收到 PC 端的数据。

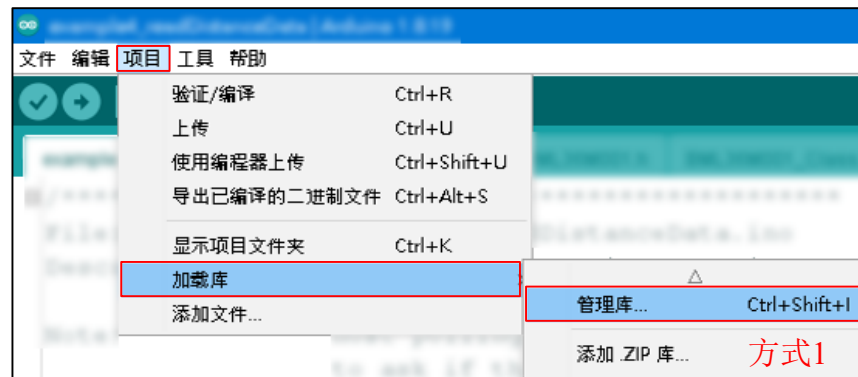
18	bool executeUpdate(void)	
	描述	更新音源
	参数	void
	返回值	音源更新情况 true: 更新完成 false: 更新失败
	备注	在 bool isUpdateBegin(void) 函数后使用 此函数会将接收到的 PC 端数据，通过 SPI 通信烧录到扩充板 此函数会执行直到音源更新完成

Arduino Lib 下载及安装

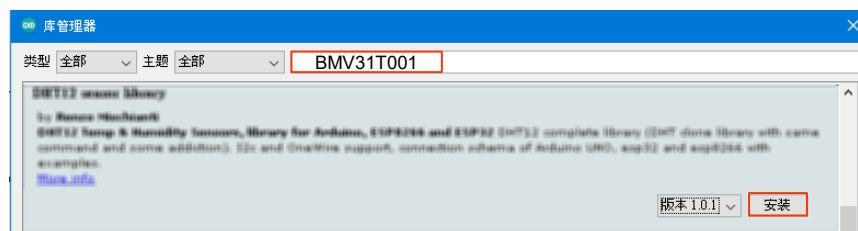
BMV31T001 Library: 可参考下面两种方法安装 BMV31T001 的 Arduino Library

方式 1: 搜索安装

搜索安装: Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BMV31T001 → 安装



搜索安装流程 1

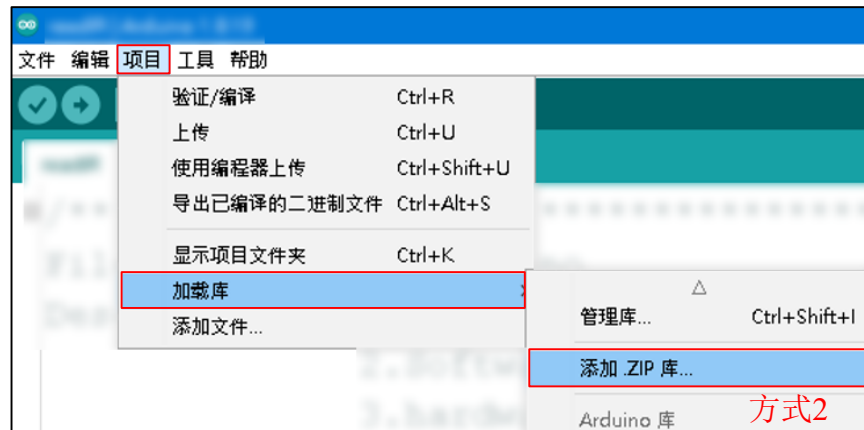


搜索安装流程 2

方式 2: 添加 .ZIP 库, 需提前下载 .ZIP 库

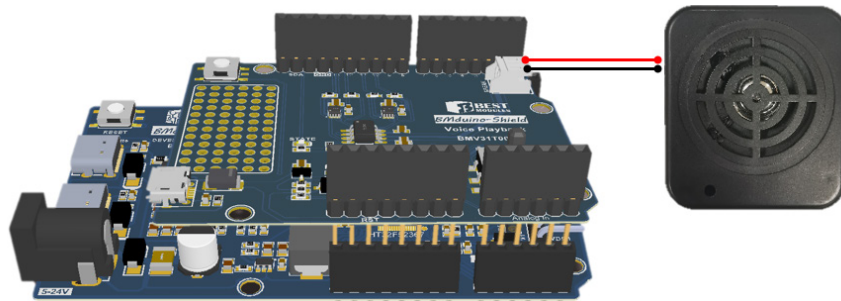
下载方法: 打开倍创官方网站 (<https://www.bestmodulescorp.com/bmv31t001.html>) 文件目录下的 Arduino 范例程式 (BMV31T001 Library)。

添加 .ZIP 库: Arduino IDE → 项目 → 加载库 → 添加 .ZIP 库 ...



Arduino 范例

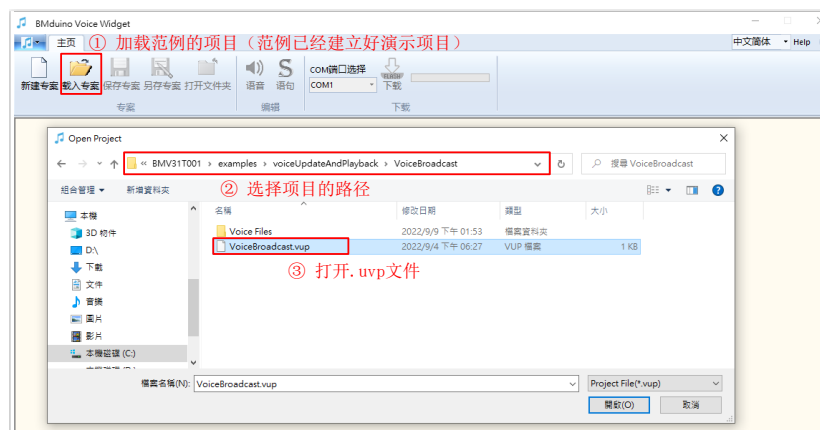
范例：voiceUpdateAndPlayback



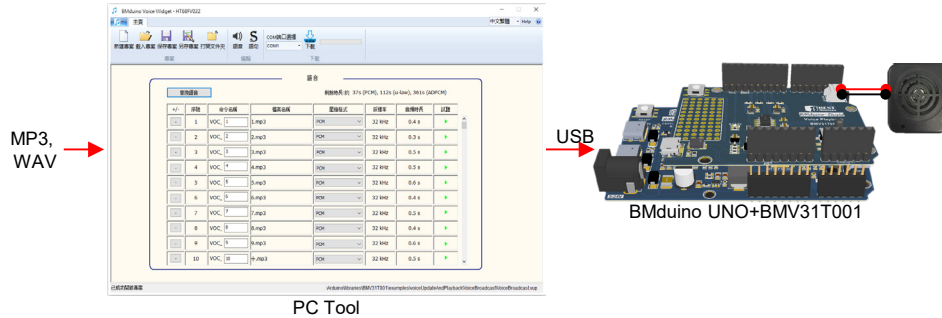
范例实现功能：在线更新音源，并可以通过按键控制播放烧录的音源。

操作步骤：

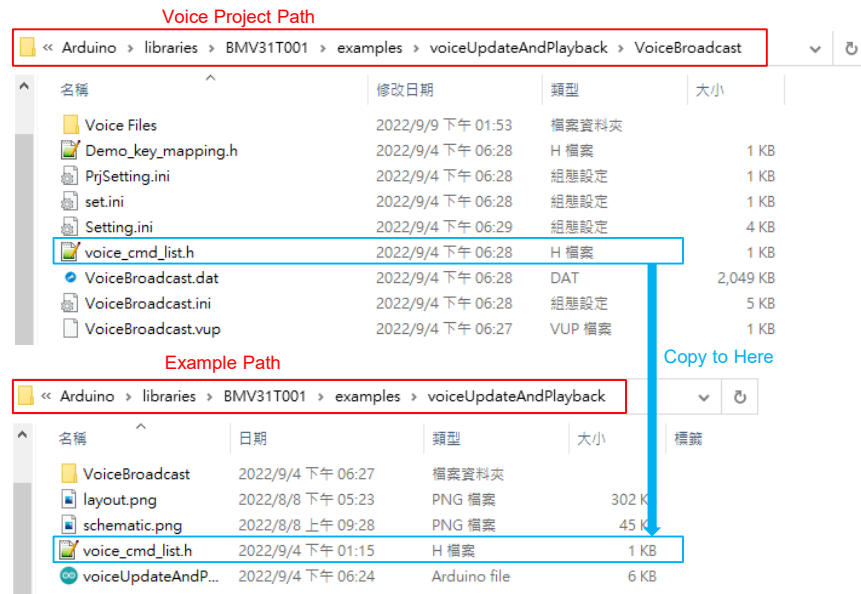
1. 范例打开方式：Arduino IDE → 文件 → 示例 → Lib 选择 (BMV31T001) → 选择范例 (voiceUpdateAndPlayback)，并下载到 BMduino UNO 开发板中。
2. 打开烧录音源小工具 BMduino Voice Widget.exe。范例已经建立好语音工程项目，直接选择范例文件夹下的 .vup，加载本范例的音源处理工程。



3. 将 USB 接入 NATIVE 标识的接口，并在 BMduino Voice Widget.exe 选择 com 口下载音源



4. 拷贝头文件到范例



5. 操作完以上步骤就可以根据按键控制语音播放，范例程序定义的按键功能如下表

按键	功能
中	播放 / 停止
上	音量 +
下	音量 -

左	上一首
右	下一首

范例详解：

- 构建对象 & 初始化扩充板

```
#include "BMV31T001.h"           // 加入扩充板
#include "voice_cmd_list.h"       // 加入平台烧录的语音信息列表
BMV31T001 myBMV31T001;          // 建立扩充板对象
#define VOICE_TOTAL_NUMBER 10    // 本次范例测试 10 个语音数据
// 从 "voice_cmd_list.h" 拿出语音名称建立语音列表
const uint8_t voice_table[VOICE_TOTAL_NUMBER]={
VOC_1,VOC_2,VOC_3,VOC_4,VOC_5,VOC_6,VOC_7,VOC_8,VOC_9,VOC_10};
uint8_t keyStatus;               // 按键的状态
uint8_t lastKeyStatus;          // 按键的状态
uint8_t playStatus;              // 当前播放状态
uint8_t playNum;                 // 当前播放曲目
#define DEFAULT_VOLUME 6         // 默认音量
uint8_t volume = DEFAULT_VOLUME; // 当前音量
uint8_t keycode = 0;             // 触发的按键编号
void setup() {
    myBMV31T001.begin();         // 初始化扩充板
    myBMV31T001.setPower(BMV31T001_POWER_ENABLE); // 使能扩充板上电
    //=====
    // 如果想要在线更新音源，请加入音源更新的串口初始化程序
    // 支持 arduino 的 serial 和 BMduinoUNO 的 USB NATIVE 接口

    myBMV31T001.initAudioUpdate();
    //=====
    delay(100);                  // 延时等待扩充板上电完成
    myBMV31T001.setVolume(DEFAULT_VOLUME); // 初始化默认音量 volume
}
```

- 在 loop 中执行音源更新、按键扫描、按键处理以及播放状态灯显示函数

```
void loop() {
    //=====
    // 如果想要在线更新音源，请加入这段程序
    if(myBMV31T001.isUpdateBegin() == BMV31T001_UPDATA_BEGIN)
    { // 检测更新信号
        myBMV31T001.executeUpdate(); // 执行音源更新
    }
    //=====
    //----- 播放指示灯控制 -----
    if(myBMV31T001.isPlaying() == BMV31T001_BUSY)
    { // 检测播放状态
        playStatus = BMV31T001_BUSY;
        myBMV31T001.setLED(BMV31T001_LED_ON); // LED 打开
    }
    else
    {
        playStatus = BMV31T001_NOBUSY;
        myBMV31T001.setLED(BMV31T001_LED_OFF); // LED 关闭
    }
}
```

```
//----- 按键扫描和处理 -----
myBMV31T001.scanKey();
if(myBMV31T001.isKeyAction() != BMV31T001_NO_KEY) // 判断是否有按键按下
{
    keyStatus = myBMV31T001.readKeyValue(); // 读取按键电平
    if(lastKeyStatus == 0) // 如果上一次按键的电平为 0
    {
        if((keyStatus & BMV31T001_KEY_MIDDLE) != 0){ // 中键按下
            keycode = BMV31T001_KEY_MIDDLE; // 记录按下的按键
        }
        else if((keyStatus & BMV31T001_KEY_UP) != 0){ // 上键按下
            keycode = BMV31T001_KEY_UP; // 记录按下的按键
        }
        else if((keyStatus & BMV31T001_KEY_DOWN) != 0){ // 下键按下
            keycode = BMV31T001_KEY_DOWN; // 记录按下的按键
        }
        else if((keyStatus & BMV31T001_KEY_LEFT) != 0){ // 左键按下
            keycode = BMV31T001_KEY_LEFT; // 记录按下的按键
        }
        else {
            keycode = BMV31T001_KEY_RIGHT; // 右键按下, 记录按下的按键
        }
    }
    lastKeyStatus = keyStatus; // 保存上次的按键状态
    switch(keycode) // 判断按下的按键做对应的动作
    {
        case BMV31T001_KEY_MIDDLE: // 中键按下
            if(playStatus == BMV31T001_BUSY) // 是否处于播放状态
            {
                myBMV31T001.playStop(); // 播放中则停止播放
            }
            else
            {
                myBMV31T001.playVoice(voice_table[playNum]); // 未在播放中
                // 则进入播放
            }
            keycode = 0;
            break;
        case BMV31T001_KEY_UP: // 上键按下
            if(volume < BMV31T001_VOLUME_MAX) // 不处于最大音量
            {
                volume++; // 音量增
                myBMV31T001.setVolume(volume);
            }
            keycode = 0;
            break;
        case BMV31T001_KEY_DOWN: // 下键按下
            if(volume > BMV31T001_VOLUME_MIN) // 不处于最小音量
            {
                volume--; // 音量减
                myBMV31T001.setVolume(volume);
            }
    }
}
```

```
        keycode = 0;
        break;
        case BMV31T001_KEY_LEFT:                // 左键按下
        if(playNum > 0)                          // 不处于第一首
        {
            playNum--;                          // 上一首
        }
        myBMV31T001.playVoice(voice_table[playNum]); // 播放
        keycode = 0;
        break;
        case BMV31T001_KEY_RIGHT:               // 右键按下
        if(playNum < VOICE_TOTAL_NUMBER - 1)    // 不处于最大音量
        {
            playNum++;                          // 下一首
        }
        myBMV31T001.playVoice(voice_table[playNum]); // 播放
        keycode = 0;
        break;
        default:
        keycode = 0;
        break;
    }
}
}
```

Copyright© 2023 by BEST MODULES CORP. All Rights Reserved.

本文件出版时倍创已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。倍创不担保任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。倍创就文中提到的信息及该信息之应用，不承担任何法律责任。此外，倍创并不推荐将倍创的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。倍创特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用倍创产品的风险完全由买方承担，如因该等使用导致倍创遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使倍创免受损害。倍创 (及其授权方，如适用) 拥有本文件所提供信息 (包括但不限于内容、数据、示例、材料、图形、商标) 的知识产权，且该信息受著作权法和其他知识产权法的保护。倍创在此并未明示或暗示授予任何知识产权。倍创拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。