

# 基于 Arduino IDE

## 智元素-格斗机器人功能函数参考文档

文档版本号	更新时间	修订人	审定人	备注
1.0	2018-06-21	刘欢		初稿
1.1	2018-09-27	刘欢		参考 修订记录
1.2	2018-11-12	刘欢		参考 修订记录

## 1. 目录

1. 目录	2
2. 用户必读:	4
2.1. 使用 Innobot 图形化编程软件进入 ArduinoIDE (推荐)	4
2.2. 使用自己安装的 ArduinoIDE 用户使用说明	7
2.2.1. 方法 1	7
2.2.2. 方法 2	8
3. 机器人控制函数	9
3.1. 函数 Begin	9
3.2. 函数 WaitBegin	9
3.3. 函数 MoveModeSet	10
3.4. 函数 MoveControl	10
3.5. 函数 RobotMode	10
3.6. 函数 MotionSet	10
3.7. 函数 MotionRun_name	11
3.8. 函数 VoicePlay	11
3.9. 函数 RobotContestState_Update	12
4. 扩展传感器操作函数	13
4.1. 函数 Begin	13
4.2. 颜色传感器	14
4.2.1. 函数 COLOR_ColorRead	14
4.2.2. 函数 COLOR_LumRead	15
4.2.3. 函数 COLOR_ReflexRead	16
4.2.4. 函数 COLOR_Read	16
4.3. 距离传感器	18
4.3.1. 函数 US_ContinuRead	18
4.3.2. 函数 US_ListenerRead	19
4.3.3. 函数 US_SingleMode	20
4.3.4. 函数 US_SingleRead	21
4.4. 温度传感器	22
4.4.1. 函数 TEMP_SetOffset	22

4.4.2. 函数 TEMP_ReadValue .....	22
4.5. 角度传感器 .....	24
4.5.1. 函数 ANGLE_AccelX .....	24
4.5.2. 函数 ANGLE_AccelY .....	24
4.5.3. 函数 ANGLE_AccelZ .....	25
4.5.4. 函数 ANGLE_GyroX .....	26
4.5.5. 函数 ANGLE_GyroY .....	26
4.5.6. 函数 ANGLE_GyroZ .....	27
4.5.7. 函数 ANGLE_Yaw .....	27
4.5.8. 函数 ANGLE_Pitch .....	28
4.5.9. 函数 ANGLE_Roll .....	28
4.5.10. 函数 ANGLE_MathHz .....	29
4.5.11. 函数 ANGLE_RTS .....	30
4.6. 触碰传感器 .....	31
4.6.1. 函数 Touch_ReadStatus .....	31
4.7. 数码管显示模块 .....	32
4.7.1. 函数 DisplayInt16 .....	32
4.7.2. 函数 DisplayFloat .....	32
4.8. 声音传感器 .....	33
4.8.1. 函数 Sound_ReadValue .....	33
5. 附录 .....	34
5.1. 语音格式处理 .....	34
5.2. 修订记录 .....	36
5.2.1. 版本 1.1 .....	36
5.2.2. 版本 1.2 .....	36
6. 疑问汇总 .....	36
6.1. 机器人不能走直线? .....	36

## 2. 用户必读:

本文档主要介绍使用 Arduino IDE 开发机器人控制程序的使用方法和函数参考。需要有一定的语言和编程基础。

如果你已经安装了本公司的 Innobot 图形化编程平台，可打开 Innobot 软件，然后启动 Arduino IDE。如果你没有安装本公司的 Innobot 图形化编程平台可以在网络上自行下载安装 Arduino IDE，然后在安装好的软件中加入本公司开发的库文件即可。

### 2.1. 使用 Innobot 图形化编程软件进入 ArduinoIDE （推荐）

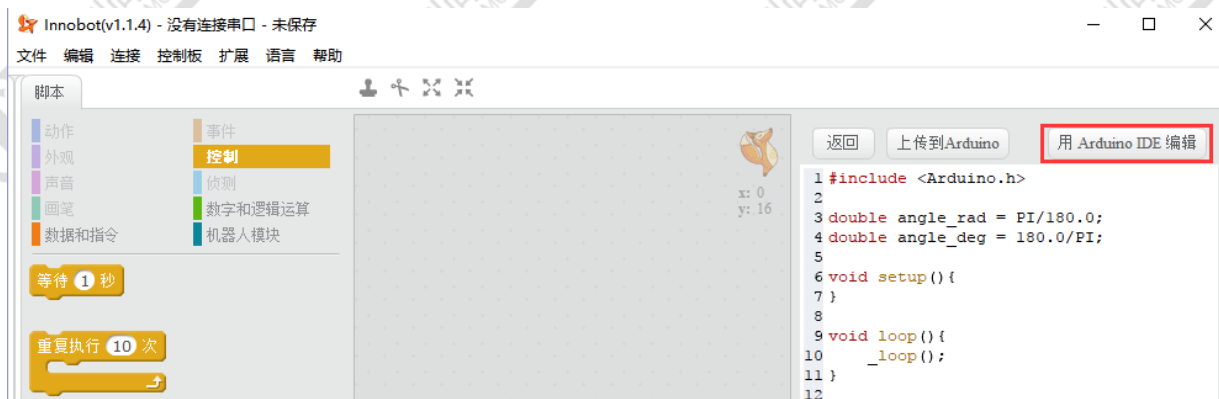
第一步：打开 Innobot 图形编程软件



第二步：进入 Arduino 模式

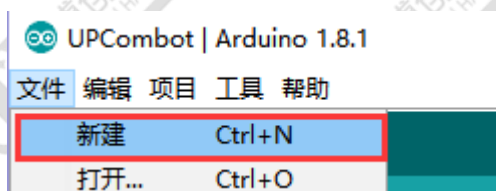


第三步：双击软件左上角的“用 ArduinoIDE 编辑”按钮，打开 ArduinoIDE

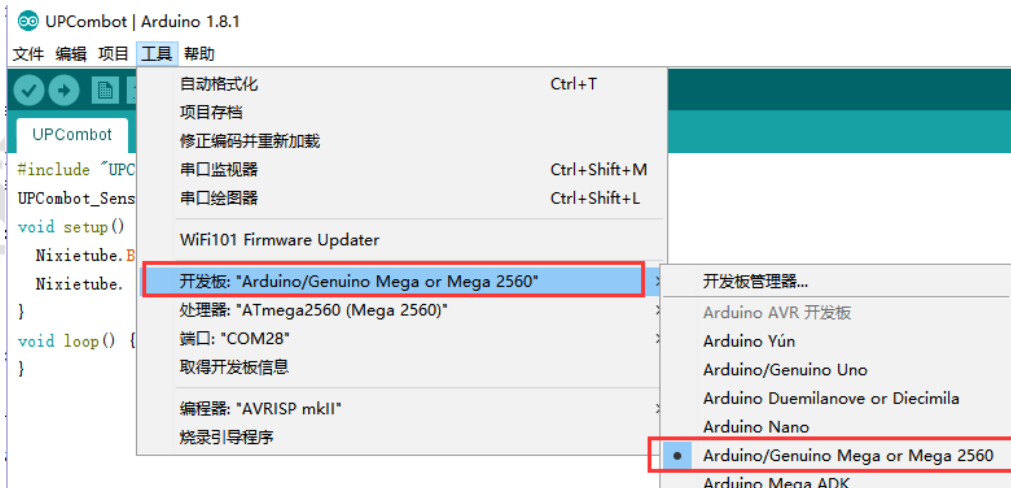


第四步：Arduino IDE 项目设置

1) 新建项目：ArduinoIDE 菜单栏->文件->新建，如图



## 2) 选择开发板: ArduinoIDE 菜单栏-&gt;工具-&gt;开发板-&gt;Arduino/Genuino Mega or Mega2560



## 3) 选择处理器: ArduinoIDE 菜单栏-&gt;工具-&gt;处理器-&gt;ATmega2560(Mega2560)



## 4) 拷贝例子代码到 ArduinoIDE 代码编辑区域



5) 将机器人的 USB 端口和电脑 USB 端口连接, 然后选择插入 USB 端口后产生的程序下载 COM 口, 当有多个 COM 端口时一定不要选错, 否则将导致程序不能下载成功。



6) COM 口选择成功后, 点击程序下载按钮, 即可下载程序, 左下角会有程序下载提示, 如果下载失败检查 COM 口是否选择错误或者程序是否编译成功。





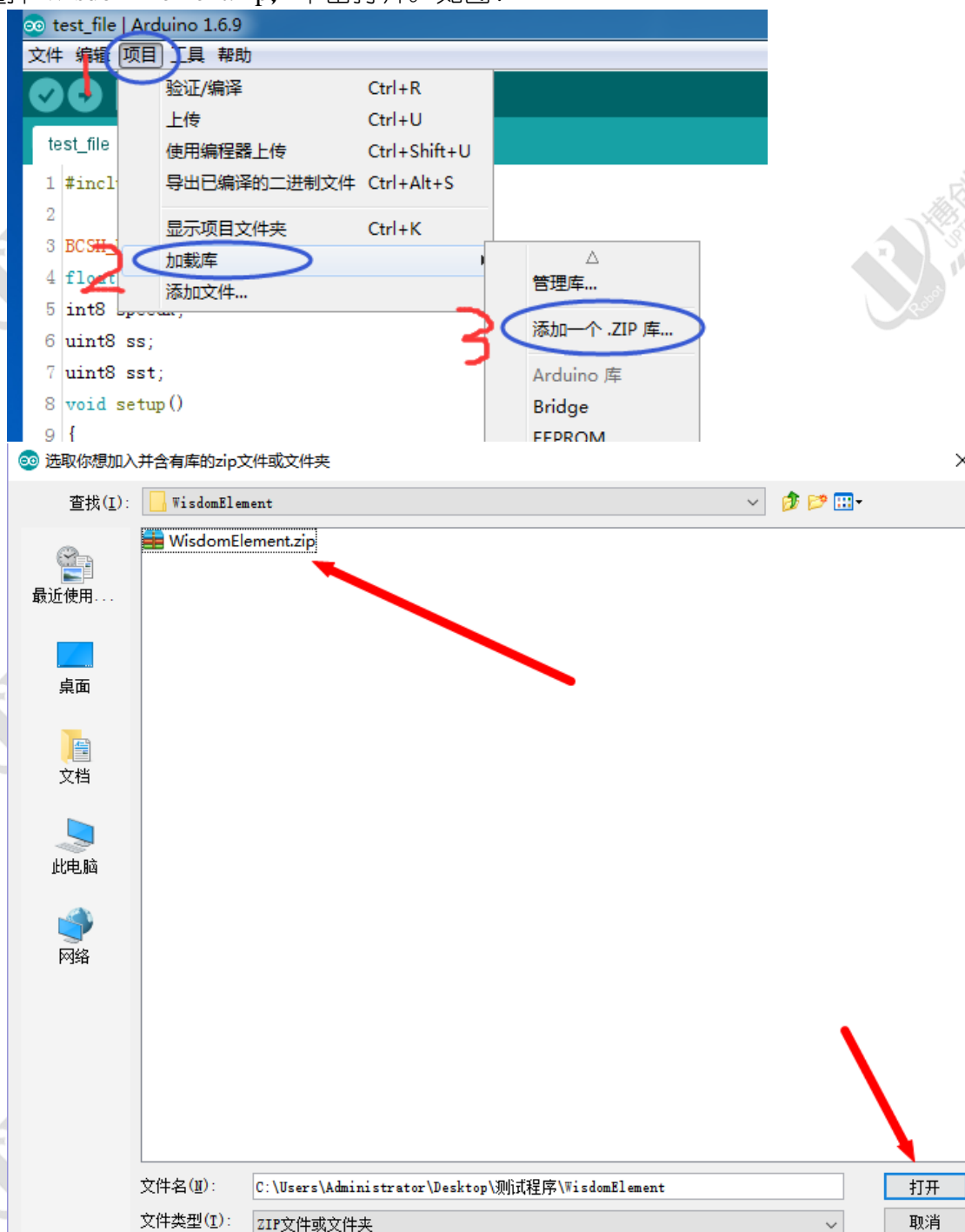
## 2.2. 使用自己安装的 ArduinoIDE 用户使用说明

自己安装 ArduinoIDE 的用户需要添加本公司的相关库文件，操作方法有 2 种，如下：  
准备工作：

- 1) 下载一个 ArduinoIDE 安装包，并安装。
- 2) 到博创尚和科技有限公司官网 [WWW.XXXX.COM](http://WWW.XXXX.COM) 下载一个 WisdomElement.zip 库文件。

### 2.2.1. 方法 1

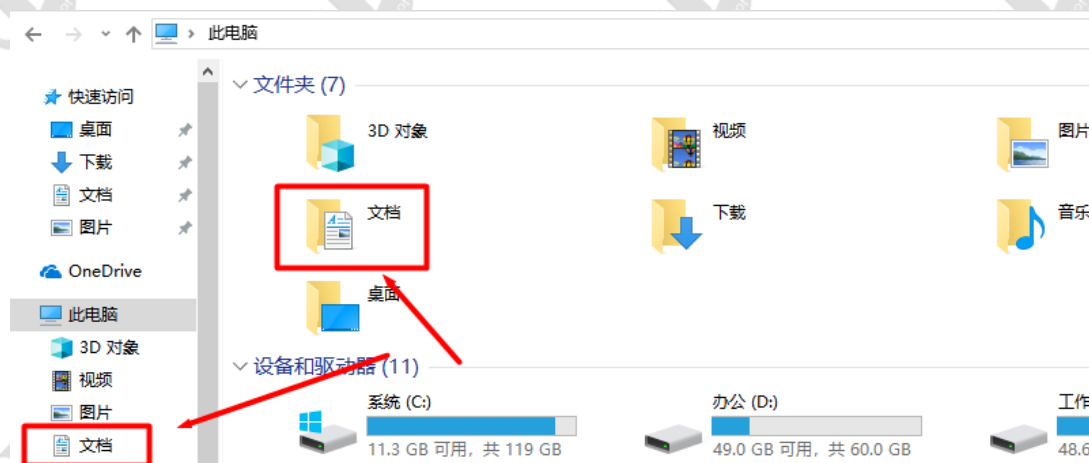
第一步：打开 ArduinoIDE，在菜单栏中依次选择：项目->加载库->添加一个.zip 库，然后选择 WisdomElement.zip，单击打开。如图：



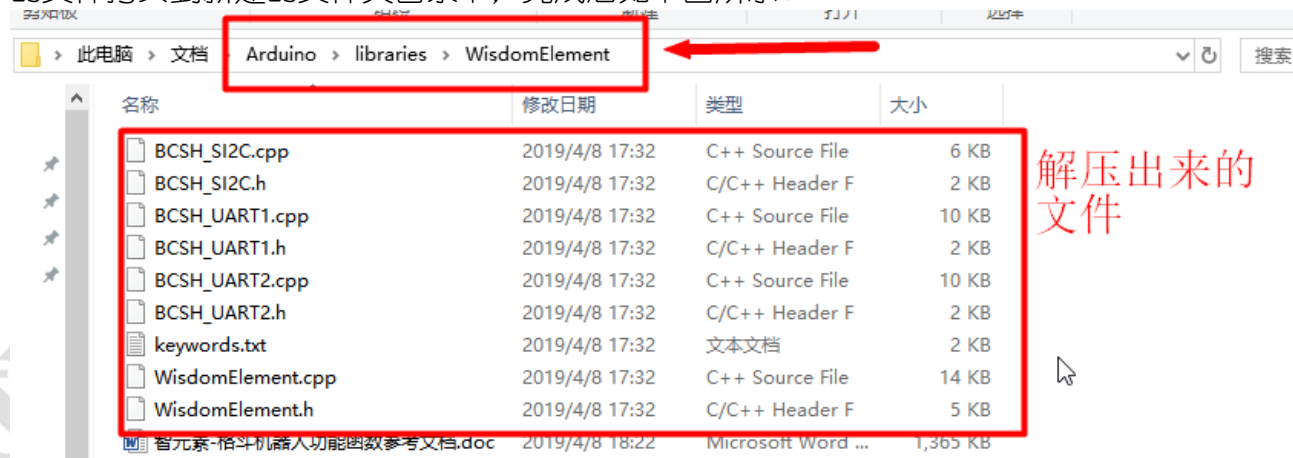
第二步：项目设置，参考 2.1 使用 Innobot 图形化编程软件进入 *ArduinoIDE* 第四步执行完上述操作后，便可以正常开发程序了。

### 2.2.2. 方法 2

安装 *ArduinoIDE* 之后可以找到目录：C:\Users\Administrator\Documents\Arduino\libraries 或者直接从磁盘我的文档进入（效果一样），如图：



其中 Administrator 为用户名，不同的用户可能不一样  
在该目录下新建一个文件夹为 *WisdomElement*，将下载好的 *WisdomElement.zip* 解压，将其中的文件拷贝到新建的文件夹目录下，完成后如下图所示：





### 3. 机器人控制函数

函数名	描述
函数 <i>Begin</i>	初始化机器人设备通信接口
函数 <i>WaitBegin</i>	初始化机器人设备通信接口，并阻塞程序直到接收到开始命令
函数 <i>MoveModeSet</i>	底盘移动模式设置
函数 <i>MoveControl</i>	底盘移动控制
函数 <i>RobotMode</i>	机器人模式设置
函数 <i>MotionSet</i>	机器人动作设置
函数 <i>MotionRun_name</i>	机器人执行指定名称动作
函数 <i>VoicePlay</i>	自定义语音播放
函数 <i>RobotContestState_Update</i>	机器人竞赛状态更新

#### 3.1. 函数 *Begin*

函数名	Begin
函数原型	void Begin()
功能描述	初始化机器人设备通信接口
输入参数	无
返回值	无
备注	如果用户希望程序烧录后不立即运行，由自己控制什么时候开始，可使用函数 <i>WaitBegin</i> 替代函数 <i>Begin</i>

[例1] 参考 [例2]

#### 3.2. 函数 *WaitBegin*

函数名	WaitBegin
函数原型	void WaitBegin ()
功能描述	初始化机器人设备通信接口，并阻塞程序直到接收到开始命令
输入参数	无
返回值	无
备注	

[例2] 初始化机器人设备，当点击开始按钮后舵机进入双闭环，向前直行一秒

```
#include "WisdomElement.h"
```

```
WisdomElement wisdom_element;
```

```
void setup(){
    wisdom_element.WaitBegin();
    wisdom_element.MoveModeSet(2);
    wisdom_element.MoveControl(0,20,0,1000);
}
```

```
void loop(){
```

}

### 3.3. 函数 MoveModeSet

函数名	MoveModeSet
函数原型	uint8 MoveModeSet(uint8 mode)
功能描述	底盘移动模式设置
输入参数	0: 单闭环（速度闭环，默认）      2: 双闭环（速度角度闭环）
返回值	略
备注	只有当底盘设置为双闭环时，底盘才能向指定的方向直行，否则底盘移动会因为障碍物或地面打滑有运行偏差

[例3] 参考 [例2]

### 3.4. 函数 MoveControl

函数名	MoveControl
函数原型	uint8 MoveControl(uint16 Angle, uint16 speed, int16 turn, uint16 time)
功能描述	底盘移动控制
输入参数	Angle: 底盘移动角度，取值 0~360，正前方为 0 度，角度逆时针递增 speed: 底盘移动速度，取值 0~100 turn: 底盘旋转速度，取值 -1000~1000，正值向左，负值向右 time: 底盘移动时间，单位毫秒
返回值	略
备注	

[例4] 参考 [例2]

### 3.5. 函数 RobotMode

函数名	RobotMode
函数原型	uint8 RobotMode(uint8 mode)
功能描述	机器人模式设置
输入参数	1: 初始模式      2: 战斗模式
返回值	略
备注	

[例5] 设置机器人进入战斗模式

```
#include "WisdomElement.h"
WisdomElement wisdom_element;
void setup(){
    wisdom_element.WaitBegin();
    wisdom_element.RobotMode(2);
}
void loop(){
}
```

### 3.6. 函数 MotionSet

函数名	MotionSet
函数原型	uint8 MotionSet(uint8 value)
功能描述	机器人动作设置
输入参数	0~3: 技能 0~3 4: 普通攻击 5: 失败动作 6: 胜利动作 7: 上电动作 8: 进入战斗模式预备动作
返回值	略
备注	

[例6] 控制机器人做出技能 1 动作

```
#include "WisdomElement.h"
WisdomElement wisdom_element;
void setup(){
    wisdom_element.WaitBegin();
    wisdom_element.MotionSet(1);
}
void loop(){
}
```

### 3.7. 函数 MotionRun\_name

函数名	MotionRun_name
函数原型	uint8 MotionRun_name(uint8 *name)
功能描述	设置机器人执行指定名称的动作
输入参数	要执行的动作名称字符串，字符串编码为 UTF8，名称最大长度不超过 63 字节
返回值	发送成功返回 1，发送失败返回 0
备注	动作名称长度或编码不对可能会导致发送失败

[例7] 控制机器人做出技能 1 动作

```
#include "WisdomElement.h"
WisdomElement wisdom_element;
void setup(){
    wisdom_element.WaitBegin();
    wisdom_element.MotionRun_name((uint8 *)"动作 2");
}
void loop(){
}
```

### 3.8. 函数 VoicePlay

函数名	VoicePlay
函数原型	uint8 VoicePlay (uint8 value)
功能描述	机器人自定义语音播放

输入参数	语音编号 0-255
返回值	略
备注	自定义语音的采样率只能为 44100HZ，8 位采样，单声道（语音处理方法参考 <a href="#">5.1 语音格式处理</a> ）。自定义语音名称长度必须为 4 位，如编号为 20 的语音，文件名称命名为 0020.wav，而且只能放置在内存中名称为 0255 的文件夹下。 注：如果语音不存在不会有任何提示

【例8】控制机器人播放自定义语音 1

```
#include "WisdomElement.h"
WisdomElement wisdom_element;
void setup(){
    wisdom_element.WaitBegin();
    wisdom_element.VoicePlay(1);
}
void loop(){
}
```

### 3.9. 函数 RobotContestState\_Update

函数名	RobotContestState_Update
函数原型	void RobotContestState_Update (uint8 state)
功能描述	机器人比赛状态更新，用于比赛状态或标志识别
输入参数	根据具体比赛要求使用
返回值	无
备注	

【例9】设置状态值为 1

```
#include "WisdomElement.h"
WisdomElement wisdom_element;
void setup(){
    wisdom_element.WaitBegin();
    wisdom_element.RobotContestState_Update (1);
}
void loop(){
}
```

## 4. 扩展传感器操作函数

所有传感器使用前必须使用 **Begin** 函数进行初始化，如果没有使用该函数则可能会导致传感器未知错误或不能正常工作。初始化实现的功能是对指定的通信接口进行初始化保证通信匹配。

表 1

函数名	描述
Begin	初始化传感器

### 4.1. 函数 Begin

表 2

函数名	Begin
函数原型	<code>void Begin(uint8 channel);</code>
功能描述	初始化数字传感器
输入参数	数字传感器的通道编号，取值 1, 2, 3, 4
返回值	无
其它	使用数字传感器之前必须调用

[例10] 初始化数字传感器，参考[例 11]或后续应用函数的例子



## 4.2. 颜色传感器

表 3

函数名	描述
<a href="#">COLOR_ColorRead</a>	读取检测到的颜色值
<a href="#">COLOR_LumRead</a>	读取环境光的亮度值
<a href="#">COLOR_ReflexRead</a>	读取反射光的强度值
<a href="#">COLOR_Read</a>	读取指定类型的返回值

### 4.2.1. 函数 COLOR\_ColorRead

表 4

函数名	COLOR_ColorRead
函数原型	uint8 COLOR_ColorRead();
功能描述	颜色模式，读取检测到的颜色值
输入参数	无
返回值	返回读取到的颜色代码，取值范围 0-6
其它	无

[例11] 读取颜色传感器颜色值，显示到数码管上

```
#include "WisdomElement.h"
```

```
UPCombot_Sensor dsensor;
```

```
UPCombot_Sensor Nixietube;
```

```
uint8 value;
```

```
void setup()
```

```
{
```

```
    dsensor.Begin(1);
```

```
    Nixietube.Begin(2);
```

```
}
```

```
void loop()
```

```
{
```

```
    value=dsensor.COLOR_ColorRead();
```

```
    Nixietube.DisplayInt16(value);
```

```
    delay(100);
```

```
}
```



#### 4.2.2. 函数 COLOR\_LumRead

表 5

函数名	COLOR_LumRead
函数原型	uint8 COLOR_LumRead();
功能描述	环境光模式，读取环境光的亮度值
输入参数	无
返回值	返回读取到的环境光亮度值，取值范围 0-100
其它	无

[例12] 读取环境光的亮度值, 显示到数码管上

```
#include "WisdomElement.h"
```

```
UPCombot_Sensor dsensor;
```

```
UPCombot_Sensor Nixietube;
```

```
uint8 value;
```

```
void setup()
```

```
{
```

```
    dsensor.Begin(1);
```

```
    Nixietube.Begin(2);
```

```
}
```

```
void loop()
```

```
{
```

```
    value=dsensor.COLOR_LumRead ();
```

```
    Nixietube.DisplayInt16(value);
```

```
    delay(100);
```

```
}
```

#### 4.2.3. 函数 COLOR\_ReflexRead

表 6

函数名	COLOR_ReflexRead
函数原型	uint8 COLOR_ReflexRead();
功能描述	反射模式，由颜色传感器主动发光，然后读取光的反射强度值
输入参数	无
返回值	反射光强度值，取值范围 0-100
其它	无

[例13] 读取光的反射强度值, 显示到数码管上

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
uint8 value;
void setup()
{
    dsensor.Begin(1);
    Nixietube.Begin(2);
}
void loop()
{
    value=dsensor. COLOR_ReflexRead ();
    Nixietube.DisplayInt16(value);
    delay(100);
}
```

#### 4.2.4. 函数 COLOR\_Read

表 7

函数名	COLOR_Read
函数原型	uint8 COLOR_Read(uint8 type);
功能描述	读取指定传感器类型下，传感器的状态值
输入参数	类型值，0（颜色模式），1（环境光模式），2（反射模式）
返回值	参考相对应模式的返回值
其它	无

[例14] 读取反射模式下，反射强度值, 显示到数码管上

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
uint8 value;
void setup()
{
    dsensor.Begin(1);
    Nixietube.Begin(2);
}
```

```
void loop()  
{  
  value=dsensor. COLOR_Read (2);  
  Nixietube.DisplayInt16(value);  
  delay(100);  
}
```

### 4.3. 距离传感器

表 8

函数名	描述
US_ContinuRead	连续测量模式数据读取
US_ListenerRead	监听模式数据读取
US_SingleMode	启动单次测量
US_SingleRead	单次测量模式数据读取

#### 4.3.1. 函数 US\_ContinuRead

表 9

函数名	US_ContinuRead
函数原型	uint16 US_ContinuRead();
功能描述	连续测量模式，连续读取测量值
输入参数	无
返回值	返回测量距离，单位为毫米，最大为 2550 毫米
其它	无

[例15] 连续读取距离传感器测量的距离值，显示到数码管上

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
uint16 value;
void setup()
{
    dsensor.Begin(1);
    Nixietube.Begin(2);
}
void loop()
{
    value=dsensor. US_ContinuRead ();
    Nixietube.DisplayInt16(value);
    delay(100);
}
```

#### 4.3.2. 函数 US\_ListenerRead

表 10

函数名	US_ListenerRead
函数原型	uint8 US_ListenerRead ();
功能描述	监听模式，被动探测周围是否有距离传感器发射超声波
输入参数	无
返回值	检测到超声波返回 1，没有检测到超声波返回 0
其它	无

[例16] 监听是否探测到超声波，将返回值显示到数码管上

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
uint8 value;
void setup()
{
    dsensor.Begin(1);
    Nixietube.Begin(2);
}
void loop()
{
    value=dsensor. US_ListenerRead ();
    Nixietube.DisplayInt16(value);
    delay(100);
}
```

### 4.3.3. 函数 US\_SingleMode

表 11

函数名	US_SingleMode
函数原型	uint8 US_SingleMode ();
功能描述	启动单次测量，调用次函数后，测量一次后停止工作
输入参数	无
返回值	成功返回 1，否则返回 0
其它	调用该函数后，至少延时 10 毫秒，才能准确得到本次测量结果

[例17] 单次模式下循环每 1 秒读取一次测量值，显示到数码管上

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
uint16 value;
void setup()
{
    dsensor.Begin(1);
    Nixietube.Begin(2);
}
void loop()
{
    dsensor.US_SingleMode ();
    delay(50);
    value=dsensor.US_SingleRead ();
    delay(1000);
    Nixietube.DisplayInt16(value);
}
```



#### 4.3.4. 函数 US\_SingleRead

表 12

函数名	US_SingleRead
函数原型	uint16 US_SingleRead ();
功能描述	单次测量模式读，读取单次测量模式测量的距离值
输入参数	无
返回值	返回测量距离，单位为毫米
其它	每次调用 US_SingleRead 函数前需要调用 US_SingleMode，否则读出来的值是上一次的测量值，如果从来都没有调用过 US_SingleMode 则读出来的值未知

[例18] 参考[例 17]

#### 4. 4. 温度传感器

表 13

函数名	描述
<a href="#">TEMP_SetOffset</a>	设置温度传感器校准值
<a href="#">TEMP_ReadValue</a>	读取温度值

##### 4. 4. 1. 函数 TEMP\_SetOffset

表 14

函数名	TEMP_SetOffset
函数原型	uint8 TEMP_SetOffset(float offset);
功能描述	设置温度校准值
输入参数	温度校准值，浮点型数据，1 位小数点精度
返回值	设置成功返回 1，否则返回 0
其它	无

[例19] 设置温度传感器校准值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
void setup()
{
    dsensor.Begin(1);
    dsensor.TEMP_SetOffset(0);
}
void loop()
{
}
```

##### 4. 4. 2. 函数 TEMP\_ReadValue

表 15

函数名	TEMP_ReadValue
函数原型	float TEMP_ReadValue();
功能描述	读取当前温度传感器的值
输入参数	无
返回值	返回温度传感器采集的温度值，1 位小数点精度，单位℃（摄氏度）
其它	无

[例20] 读取温度传感器值，并显示到数码管上

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
float value;
void setup()
{
    dsensor.Begin(1);
    Nixietube.Begin(2);
}
```

```
}  
void loop()  
{  
  value=dsensor.  TEMP_ReadValue ();  
  delay(1000);  
  Nixietube.DisplayFloat(value,1);  
}
```

## 4.5. 角度传感器

表 16

函数名	描述
<a href="#">ANGLE_AccelX</a>	读取 X 轴加速度值
<a href="#">ANGLE_AccelY</a>	读取 Y 轴加速度值
<a href="#">ANGLE_AccelZ</a>	读取 Z 轴加速度值
<a href="#">ANGLE_GyroX</a>	读取 X 轴倾角值
<a href="#">ANGLE_GyroY</a>	读取 Y 轴倾角值
<a href="#">ANGLE_GyroZ</a>	读取 Z 轴倾角值
<a href="#">ANGLE_Yaw</a>	读取角度传感器偏航值
<a href="#">ANGLE_Pitch</a>	读取角度传感器俯仰值
<a href="#">ANGLE_Roll</a>	读取角度传感器横滚值
<a href="#">ANGLE_MathHz</a>	读取解算速率
<a href="#">ANGLE_RTS</a>	重新校准角度传感器

### 4.5.1. 函数 ANGLE\_AccelX

表 17

函数名	ANGLE_AccelX
函数原型	int16 ANGLE_AccelX();
功能描述	读取 X 轴加速度值
输入参数	无
返回值	X 轴加速度值
其它	无

**[例21]** 读取 X 轴加速度值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_AccelX ();
    Nixietube.DisplayInt16(value);
    delay(100);
}
```

### 4.5.2. 函数 ANGLE\_AccelY

表 18

函数名	ANGLE_AccelY
-----	--------------

函数原型	int16 ANGLE_AccelY();
功能描述	读取 Y 轴加速度值
输入参数	无
返回值	Y 轴加速度值
其它	无

**[例22]** 读取 X 轴加速度值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_AccelY ();
    Nixietube.DisplayInt16(value);
    delay(100);
}
```

#### 4.5.3. 函数 ANGLE\_AccelZ

表 19

函数名	ANGLE_AccelZ
函数原型	int16 ANGLE_AccelZ();
功能描述	读取 Z 轴加速度值
输入参数	无
返回值	Y 轴加速度值
其它	无

**[例23]** 读取 Z 轴加速度值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_AccelZ ();
    Nixietube.DisplayInt16(value);
    delay(100);
}
```

}

#### 4.5.4. 函数 ANGLE\_GyroX

表 20

函数名	ANGLE_GyroX
函数原型	int16 ANGLE_GyroX();
功能描述	读取 X 轴角速度值
输入参数	无
返回值	
其它	无

[例24] 读取 X 轴角速度值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_AccelX ();
    Nixietube.DisplayInt16(value);
    delay(1000);
}
```

#### 4.5.5. 函数 ANGLE\_GyroY

表 21

函数名	ANGLE_GyroY
函数原型	int16 ANGLE_GyroY();
功能描述	读取 Y 轴角速度值
输入参数	无
返回值	
其它	无

[例25] 读取 Y 轴角速度值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
```



```
void loop()
{
    value=dsensor. ANGLE_AccelY ();
    Nixietube.DisplayInt16(value);
    delay(1000);
}
```

#### 4.5.6. 函数 ANGLE\_GyroZ

表 22

函数名	ANGLE_GyroZ
函数原型	int16 ANGLE_GyroZ();
功能描述	读取 Z 轴角速度值
输入参数	无
返回值	
其它	无

**[例26]** 读取 Z 轴角速度值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_AccelZ ();
    Nixietube.DisplayInt16(value);
    delay(1000);
}
```

#### 4.5.7. 函数 ANGLE\_Yaw

表 23

函数名	ANGLE_Yaw
函数原型	uint ANGLE_Yaw();
功能描述	读取角度传感器偏航值
输入参数	无
返回值	偏航角度，单位度
其它	返回值扩大了 10 倍，比如 40.1 度，实际返回 401

**[例27]** 读取角度传感器偏航值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
```

```
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_Yaw ();
    Nixietube.DisplayInt16(value);
    delay(1000);
}
```

#### 4.5.8. 函数 ANGLE\_Pitch

表 24

函数名	ANGLE_Pitch
函数原型	int16 ANGLE_Pitch ();
功能描述	读取角度传感器俯仰值
输入参数	无
返回值	角度传感器俯仰值，单位度
其它	返回值扩大了 10 倍，比如 40.1 度，实际返回 401

**[例28]** 读取角度传感器俯仰值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_Pitch ();
    Nixietube.DisplayInt16(value);
    delay(1000);
}
```

#### 4.5.9. 函数 ANGLE\_Roll

表 25

函数名	ANGLE_Roll
函数原型	uint ANGLE_Roll ();
功能描述	读取角度传感器滚转值

输入参数	无
返回值	角度传感器滚转值，单位度
其它	返回值扩大了 10 倍，比如 40.1 度，实际返回 401

**[例29]** 读取角度传感器滚转值

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_Roll ();
    Nixietube.DisplayInt16(value);
    delay(1000);
}
```

#### 4.5.10. 函数 ANGLE\_MathHz

表 26

函数名	ANGLE_MathHz
函数原型	int16 ANGLE_MathHz ();
功能描述	读取角度传感器解算速率
输入参数	无
返回值	角度传感器解算速率，单位 HZ
其它	无

**[例30]** 读取角度传感器解算速率

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
UPCombot_Sensor Nixietube;
int16 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=dsensor. ANGLE_MathHz ();
    Nixietube.DisplayInt16(value);
    delay(1000);
}
```

## 4.5.11. 函数 ANGLE\_RTS

表 27

函数名	ANGLE_RTS
函数原型	uint8 ANGLE_RTS();
功能描述	重新校准角度传感器
输入参数	无
返回值	复位成功返回 1，否则返回 0
其它	无

[例31] 重新校准角度传感器

```
#include "WisdomElement.h"
UPCombot_Sensor dsensor;
void setup()
{
    dsensor.Begin(1);
    dsensor.ANGLE_RTS();
}
void loop()
{
}
```

## 4. 6. 触碰传感器

表 28

函数名	描述
Touch_ReadStatus	读取触碰传感器状态

### 4. 6. 1. 函数 Touch\_ReadStatus

表 29

函数名	Touch_ReadStatus
函数原型	uint8 Touch_ReadStatus();
功能描述	读取触碰传感器状态
输入参数	无
返回值	触碰到物体返回 1，否则返回 0
其它	无

[例32] 读取触碰传感器状态，显示到数码管上

```
#include "WisdomElement.h"
BCSH_ASensor asensor;
UPCombot_Sensor Nixietube;
uint8 value;
void setup()
{
    Nixietube.Begin(1);
    dsensor.Begin(2);
}
void loop()
{
    value=asensor.Touch_ReadStatus ();
    Nixietube.DisplayInt16(value);
    delay(100);
}
```

## 4.7. 数码管显示模块

函数名	描述
DisplayInt16	数码管显示一个有符号整数
DisplayFloat	设置数码管显示一个小数

### 4.7.1. 函数 DisplayInt16

表 30

函数名	DisplayInt16
函数原型	void DisplayInt16(int16 value)
功能描述	数码管显示一个有符号整数
输入参数	整数参数范围-32768~+32767
返回值	无
其它	无

**[例33]** 设置数码管显示 1234

```
#include "WisdomElement.h"
UPCombot_Sensor Nixietube;
void setup() {
    Nixietube.Begin(1);
    Nixietube.DisplayInt16(1234);
}
void loop() {
}
```

### 4.7.2. 函数 DisplayFloat

表 31

函数名	DisplayFloat
函数原型	void DisplayFloat(float value,uint8 point)
功能描述	设置数码管显示一个小数
输入参数	value: 显示小数数值 point: 显示的小数点位数, 取值范围 0-3
返回值	无
其它	无

**[例34]** 设置数码管显示 45.32

```
#include "WisdomElement.h"
UPCombot_Sensor Nixietube;
void setup() {
    Nixietube.Begin(1);
    Nixietube.DisplayFloat(45.32,2);
}
void loop() {
}
```



## 4.8. 声音传感器

表 32

函数名	描述
Sound_ReadValue	读取声音传感器值

### 4.8.1. 函数 Sound\_ReadValue

表 33

函数名	Sound_ReadValue
函数原型	uint16 Sound_ReadValue();
功能描述	读取声音传感器值
输入参数	无
返回值	返回一个值，指示音量的大小，取值范围 0-100，声音越大值越大。
其它	无

[例35] 读取声音传感器值，并通过串口输出

```
#include "WisdomElement.h"
```

```
WisdomElement_Sensor sensor;
```

```
uint16 value;
```

```
void setup()
```

```
{
```

```
    sensor.Begin(1);
```

```
    Serial.begin(115200);
```

```
}
```

```
void loop()
```

```
{
```

```
    value=sensor.Sound_ReadValue();
```

```
    Serial.print(value);
```

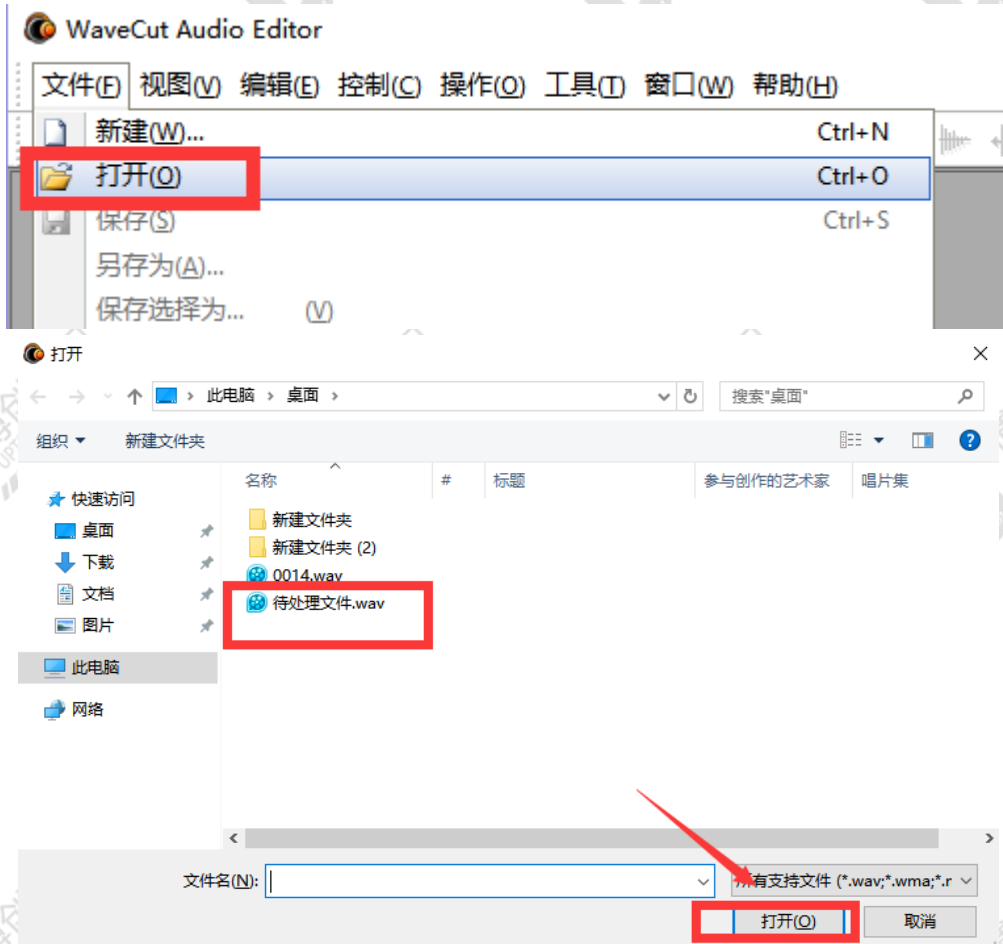
```
    Serial.print("\r\n");
```

```
}
```

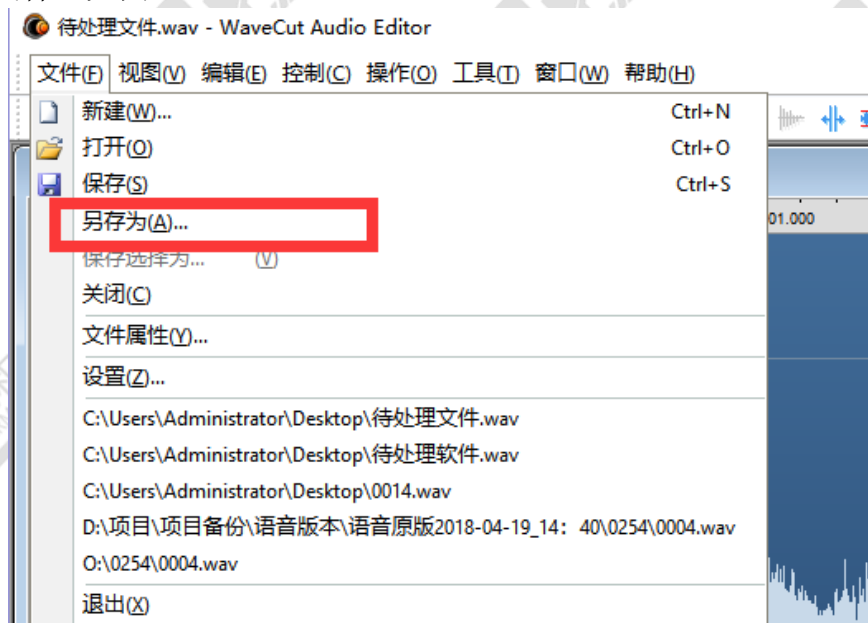
## 5. 附录

### 5.1. 语音格式处理

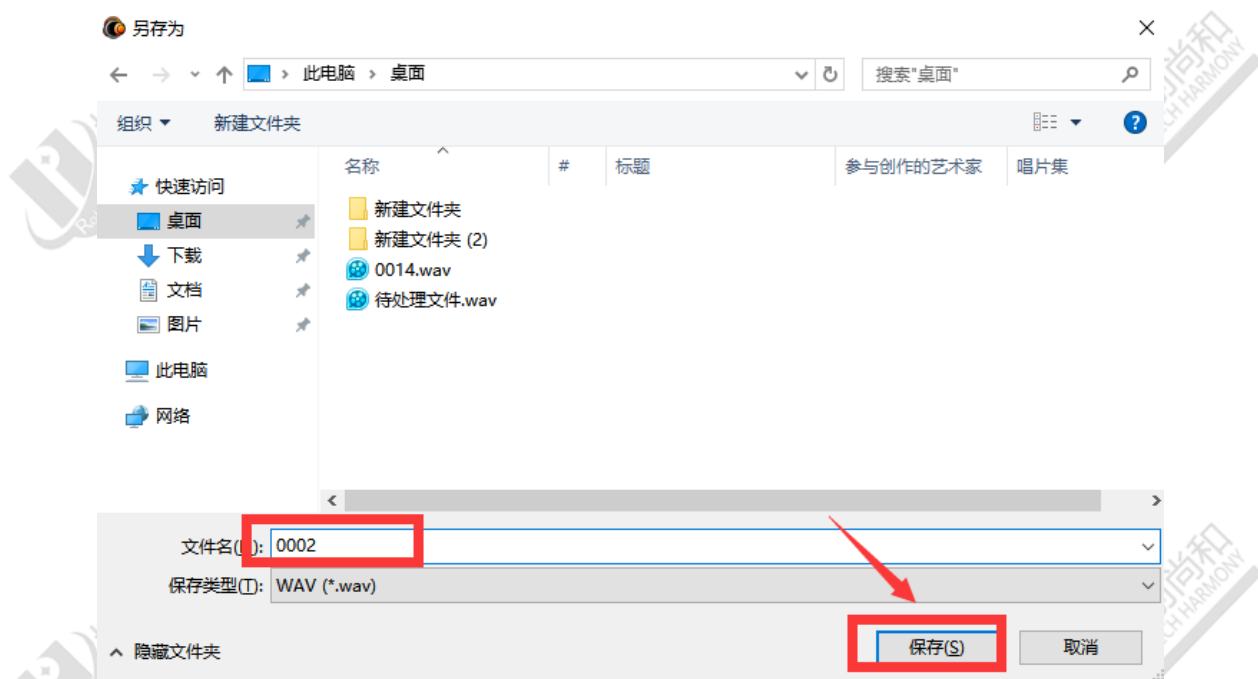
第一步：使用 wavecut 软件或其它功能类似软件，打开你的音频文件，如图：



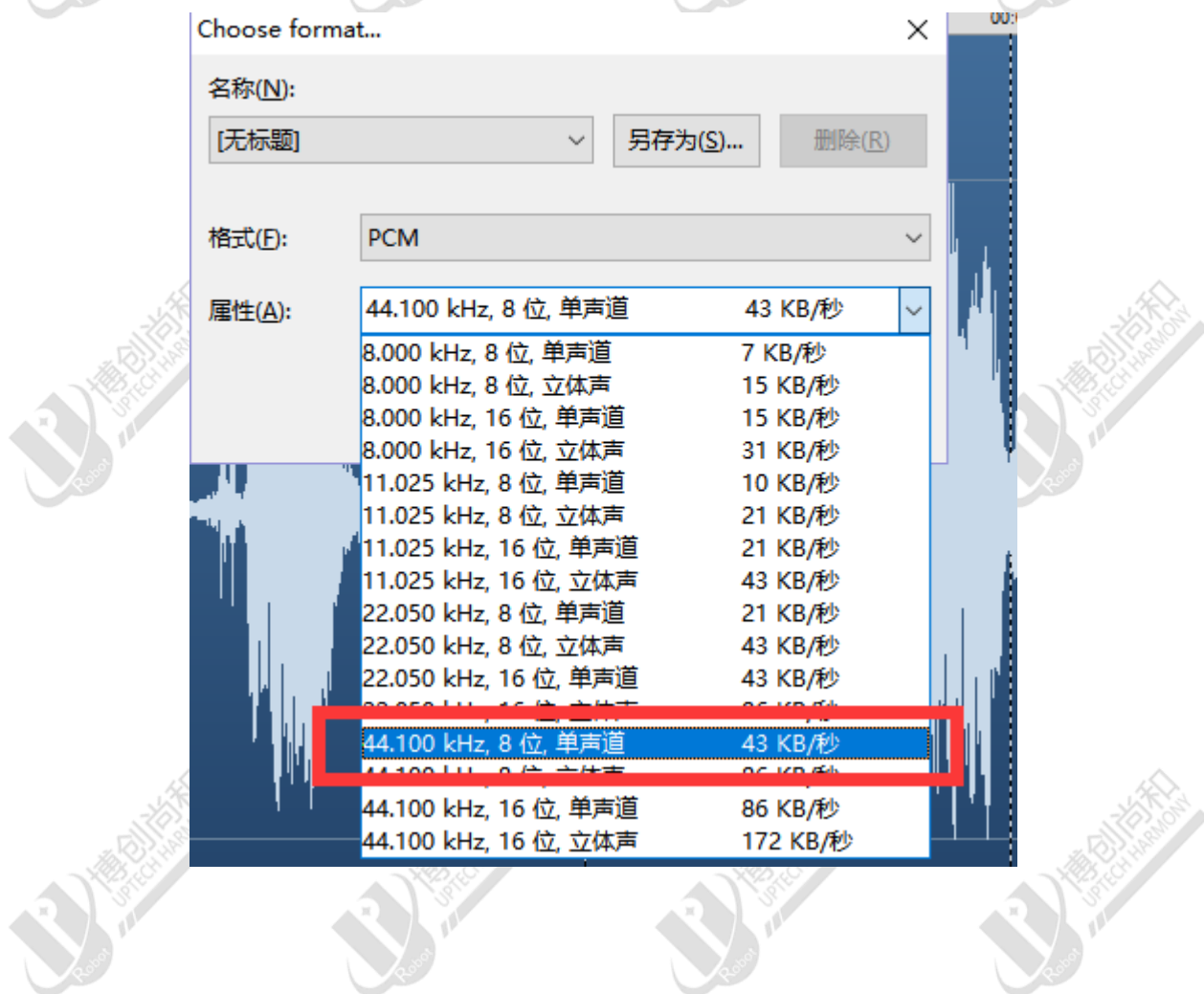
第二步：另存文件，如图



第三步：设置为符合规范的名称



第四步：选择合适的采样率以及编码格式，然后单击确定即可



## 5.2. 修订记录

### 5.2.1. 版本 1.1

1. 超声传感器更名为距离传感器

### 5.2.2. 版本 1.2

1. 增加声音传感器

## 6. 疑问汇总

### 6.1. 机器人不能走直线？

1. 机器人需要进入双闭环模式才能直线运行。调用 [函数 MoveModeSet](#)