

ViusalHMI - 音频播放

在HMI（人机界面）系统中，支持通过集成Lua脚本调用指定路径的音频文件，实现灵活、可编程的音频播放功能，用于提示、操作反馈等交互场景。

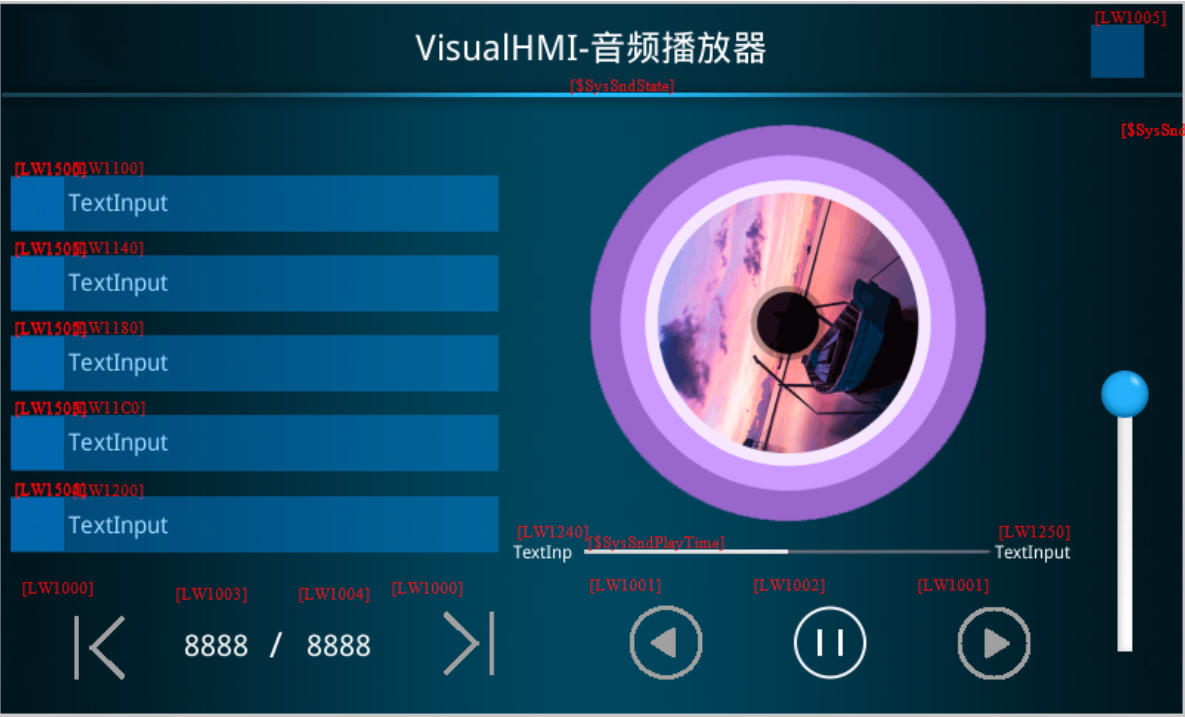
适用范围：VisualHMI - HMI&M系列&Dx系列

例程下载链接：[ViusalHMI - 音频播放\(点击下载\)](#)

1.画面配置

涉及的功能有歌曲列表翻页、歌曲上下首切换、歌曲播放状态（时间进度、播放状态）、音量控制等，所添加的控件地址和用途如表 4-1所示，画面配置如图 4-2所示

控件类型	地址	用途
字状态按钮	LW1000→设置0	歌曲列表向左翻页，LW1006=0不允许点击
字状态按钮	LW1000→设置1	歌曲列表向右翻页 LW1005=0不允许点击
字状态按钮	LW1001→设置0	上一首 LW1005=0不允许点击
字状态按钮	LW1001→设置1	下一首 LW1005=0不允许点击
位状态指示灯	LW1002	0-暂停，1-继续播放 LW1005=0不允许点击
数值	LW1003	显示歌曲列表当前页
数值	LW1004	显示歌曲列表总页数
位状态指示灯	LW1005	1-SD根目录有歌曲
文本	LW1100~1200，128个字节	显示歌曲名称
文本	LW1240	歌曲播放的当前时间 LW1005=0隐藏
文本	LW1250	歌曲播放的总时间 LW1005=0隐藏
进度条、滑块	LW0140→系统变量	喇叭音量调节
播放进度条	LW0143 →系统变量	歌曲播放的当前时间 LW1005=0隐藏
播放进度条	LW1006	进度条，数值范围，最小值
播放进度条	LW0144→系统变量	进度条，数值范围，最大值
GIF	LW0142 →系统变量	0x80 播放中



2.Lua脚本

2.1.API说明

2.1.1.play_sound(filename)

播放音频文件函数，`play_sound(filename)` 是 HMI 系统提供的**音频播放接口**，用于在运行时**触发指定音频文件的播放**。该功能常用于操作反馈、告警提示、语音引导等场景，提升人机交互的直观性与用户体验。

参数说明

参数	类型	说明
<code>filename</code>	string	音频文件路径：绝对路径

2.1.2.stop_sound()

停止音频播放函数，`stop_sound()` 是 HMI 系统提供的**音频控制接口**，用于**立即终止当前正在播放的音频文件**

2.1.3.pause_sound()

暂停音频播放函数，`pause_sound()` 是 HMI 系统提供的**音频播放控制接口**，用于**临时暂停当前正在播放的音频**（由 `play_sound` 启动），并在后续通过 `resume_sound()` 恢复播放。

2.1.4.resume_sound()

恢复音频播放函数，`resume_sound()` 是 HMI 系统提供的音频播放控制接口，用于恢复由 `pause_sound()` 暂停的音频播放，从暂停时的位置继续播放。该函数必须与 `pause_sound()` 配套使用。

2.2.程序

2.2.1遍历MP3文件

```
-- 获取扩展名
-- 自定义函数（非回调函数，可以修改）
-- @str 需要获取后缀名的字符串
function get_extension(str)

    local file_type = ''
    local token_s, token_e = string.find(str, "%.", 1)

    if token_s ~= nil
    then
        file_type = string.sub(str, (token_s + 1), string.len(str))
    end

    return file_type
end

--@path      :文件路径
--@filename:文件名称
--@type      :0 文件夹, 1 文件
--@fsize     :文件大小
function on_list_dir(path, filename, type, fsize)

    if type == 1
    then
        local cur_file_type = get_extension(filename)
        print('filename = '..filename)
        if cur_file_type == 'mp3'
        then
            music_cnt = music_cnt + 1
            musicNameListTb[music_cnt] = path..'/'..'filename
            print(' music_cnt = '..music_cnt..' / path = '..
musicNameListTb[music_cnt])
        end
    end

end

end

-- 系统回调函数，插入SD卡自动回调
function on_sd_inserted(dir)

    sd_dir = dir
```

```

list_dir(sd_dir)

if music_cnt > 0
then

    music_allpage = math.modf(music_cnt / 5)

    if music_cnt % 5 > 0
    then
        music_allpage = music_allpage + 1
    end

    set_uint16(VT_LW, 0x1003, 1)
    set_uint16(VT_LW, 0x1004, music_allpage)
    set_uint16(VT_LW, 0x1005, 1)
    updatMisucList(music_curpage)
    updatMisucNameColor(0)
end
end

-- 系统回调函数，拔掉SD卡自动回调
function on_sd_removed()

    --复位清除相关变量
    musicNameListTb      = {}
    music_cnt            = 0
    music_curpage        = 1
    music_allpage        = 1
    curPalyIindex        = 0

    set_uint16(VT_LW, 0x1003, 0)
    set_uint16(VT_LW, 0x1004, 0)
    set_uint16(VT_LW, 0x1005, 0)

    for i = 1, 5
    do
        set_string(VT_LW, 0x1100 + (i - 1)*(0x40), '')
        set_uint16(VT_LW, 0x1500 + (i - 1)*1, 0)
    end

end
end

```

2.2.2.显示列表

```

--刷新音乐名称列表
function updatMisucNameColor(index)

    if index > 0
    then
        local line = index % 5

        if line == 0
        then
            line = 5
        end
    end
end

```

```

end

for i = 1, 5
do
    if i == line
    then
        wgt_set_fcolor(sc_music, i, play_color)
    else
        wgt_set_fcolor(sc_music, i, noplay_color)
    end
end
end
else
for i = 1, 5
do
    wgt_set_fcolor(sc_music, i , noplay_color)
end
end
end
end

```

```

function on_update(slave,vtype,addr)

if VT_LW == vtype
then
    if addr == 0x1000 --翻页
    then
        local val      = get_uint16(VT_LW, 0x1000)
        local cur_page = get_uint16(VT_LW, 0x1003)
        local all_page = get_uint16(VT_LW, 0x1004)

        if val == 0    --上一页
        then
            cur_page = cur_page - 1
            if cur_page <= 1
            then
                cur_page = 1
            end
            set_uint16(VT_LW, 0x1003, cur_page)
            updatMisucList(cur_page)

        elseif val == 1    --下一页
        then
            cur_page = cur_page + 1
            if cur_page > all_page
            then
                cur_page = all_page
            end
            set_uint16(VT_LW, 0x1003, cur_page)
            updatMisucList(cur_page)
        end
    ....

end

```

```
end
end
```

2.2.3.播放控制

历程可以触控点击上一首、下一首、列表的某一个音频播放；同时播放完可以自动播放下一个音频。点击播放触发on_update(...), 上一首/下一首通过全局变量curPalyIindex 自加或自减, 获取musicNameListTb缓存中的第几个视频；暂停/继续, 通过调用play_sound()、resume_sound()控制。自动播放下一个音频, 播放过程通过on_run(), 在updatMisucPlayState()循环检测播放状态, 通过curPalyIindex 自加, 计算下一个音频的路径。代码如下所示

```
--刷新播放状态、检测播放完自动播放下一首
function updatMisucPlayState()

    local SysSndState      = get_uint16(VT_LW, 0x0142)    --获取播放状态
    local SysSndPlayTime   = get_uint16(VT_LW, 0x0143)    --获取当前播放的时间
    local SysSndTotalTime  = get_uint16(VT_LW, 0x0144)    --获取当前播放的总时间

    set_string(VT_LW, 0x1240, string.format('%02d', (SysSndPlayTime //
60))..' ':'..'..(string.format('%02d', (SysSndPlayTime %60))))    --显示播放的当前的时
间"xx:xx"
    set_string(VT_LW, 0x1250, string.format('%02d', (SysSndTotalTime //
60))..' ':'..'..(string.format('%02d', (SysSndTotalTime %60))))    --显示播放的总时
间"xx:xx"

    if ((SysSndState >> 6) & 0x01) == 0x01
    then
        curPalyIindex = curPalyIindex + 1 --下一首索引+1
        if curPalyIindex > music_cnt      --大于歌曲总数,播放第1首
        then
            curPalyIindex = 1
        end

        local cur_play_page = math.modf(curPalyIindex / 5)
        if curPalyIindex % 5 > 0
        then
            cur_play_page = cur_play_page + 1
        end

        local cur_page = get_uint16(VT_LW, 0x1003)
        if cur_play_page == cur_page
        then
            updatMisucNameColor(curPalyIindex)
        else
            updatMisucNameColor(0)
        end

        play_sound(musicNameListTb[curPalyIindex])
        SysSndState = set_valbit(SysSndState, 6, 0) --bit6 置为0
        set_uint16(VT_LW, 0x0142, SysSndState)    --复位播放状态
    end
    redraw()
end
```

```

function on_run(screen)
    updatMisucPlayState()
end

function on_update(slave,vtype,addr)

    if VT_LW == vtype
    then
        .....

    elseif addr == 0x1505    --点击列表播放
    then
        local val = get_uint16(VT_LW, 0x1505)
        local cur_page = get_uint16(VT_LW, 0x1003)
        local temp = curPalyIindex
        set_uint16(VT_LW, 0x1002, 1) --设置播放按钮

        curPalyIindex = (cur_page - 1) * 5 + val
        if curPalyIindex <= music_cnt
        then
            if string.len(musicNameListTb[curPalyIindex]) > 0
            then
                print('.... curPalyIindex = '..curPalyIindex)
                updatMisucNameColor(curPalyIindex)
                if curPalyIindex <= music_cnt
                then
                    stop_sound()
                    play_sound(musicNameListTb[curPalyIindex])
                end
            end
        else
            curPalyIindex = temp
        end

    elseif addr == 0x1001
    then
        local val = get_uint16(VT_LW, 0x1001)
        set_uint16(VT_LW, 0x1002, 1) --设置播放按钮

        if val == 0    --上一首
        then
            curPalyIindex = curPalyIindex - 1
            if curPalyIindex < 1
            then
                curPalyIindex = music_cnt
            end

        elseif val == 1    --下一首
        then
            curPalyIindex = curPalyIindex + 1
            if curPalyIindex > music_cnt
            then
                curPalyIindex = 1
            end
        end
    end
end

```

```

        end
    end

    --刷新当前页
    local cur_play_page = math.modf(curPalyIindex / 5)
    if curPalyIindex % 5 > 0
    then
        cur_play_page = cur_play_page + 1
    end
    set_uint16(VT_LW, 0x1003, cur_play_page)
    play_sound(musicNameListTb[curPalyIindex])
    updatMisucList(cur_play_page)
    updatMisucNameColor(curPalyIindex)

elseif addr == 0x1002 --播放按钮
then
    local val = get_uint16(VT_LW, 0x1002)
    if val == 0
    then
        pause_sound()

        elseif val == 1
        then
            resume_sound()
            --play_sound(musicNameListTb[curPalyIindex])
        end
    end
end
end
end
end

```

2.3.运行预览

编译下载后，在实体屏测试，在SD卡放.mp3格式音频文件，插入实体屏，进行测试