

JQ8X00_UART 库文件使用说明

```
/*
*版权信息：深圳市佳仁科技有限公司
*当前版本：V1.0
*MCU 型号：Arduino UNO
*开发环境：Arduino 1.8.12
*完成日期：2020-06-27
*作 者：JW
*程序功能：JQ8X00 系列例程
*/
```

一：库文件使用注意事项

- (1) JQ8X00_UART 库文件必须对串口做初始化配置，串口配置：Serial.begin(9600);
- (2) 库函数使用前需实例化一个对象，实例化的同时需要初始化对象：
JQ8X00(uint8_t IO_bussycheck, uint8_t bussychecken);
IO_bussycheck：忙检测 IO 口，bussychecken:忙检测使能
例如：JQ8X00 JQ8X00_Device(2, 0);其中 JQ8X00 是类，JQ8X00_Device 是实例化的对象，配置一个实例：忙检测 IO 编号为 2，不开启忙检测。对象名字可以根据自己更改。
- (3) 库文件需要放入 Arduino 安装目录下的 libraries 文件夹。

二：API 函数

库文件提供 4 个 API 函数供用户调用，分别为：

- void Command(UartCommand Command);
- void Command_Data(UartCommandData Command, uint8_t DATA);
- void AppointPlay(JQ8X00_Symbol MODE, char *DATA);
- void MakeUpPlay(unsigned char *DATA, unsigned char Len);

(1) void Command(UartCommand Command);

指令通信方式

UartCommand 为枚举类型，Command 可供选择的参数如下：

typedef enum

```
{
CheckPlayState = 0x01,      /*查询播报状态*/
Play = 0x02,                /*播放*/
pause = 0x03,               /*暂停*/
```

深圳市佳仁科技有限公司 地址：深圳市龙岗区布吉街道京南路锦龙大厦 8 楼

```

Stop = 0x04,          /*停止*/
LastSong = 0x05,      /*上一曲*/
NextSong = 0x06,      /*下一曲*/
CheckOnlineDiskSign = 0x09, /*查询当前在线盘符*/
CheckCurrentDiskSign = 0x0A, /*查询当前播放盘符*/
CheckTotalTrack = 0x0C, /*查询总曲目*/
CurrentTrack = 0x0D,   /*当前曲目*/
LastFloder = 0x0E,     /*上一个文件夹目录*/
NextFloder = 0x0F,     /*下一个文件夹目录*/
EndPlay = 0x10,        /*结束播放*/
CheckFloderFirstTrack = 0x11, /*查询文件目录首曲目*/
CheckFloderAllTrack = 0x12, /*查询文件目录总曲目*/
AddVolume = 0x14,      /*音量加*/
DecVolume = 0x15,      /*音量减*/
EndZHPlay = 0x1C,      /*结束组合播报*/
CheckSongShortName = 0x1E, /*查询歌曲短文件名*/
EndLoop = 0x21,        /*结束复读*/
GetTotalSongTime = 0x24, /*获取当前曲目总时间*/
OpenPlayTime = 0x25,   /*播放时间开发送*/
ClosePlayTime = 0x26,  /*关闭播放时间发送*/
} UartCommand;          //无数据的指令,起始码-指令类型-数据长度
-校验和
例如, 播放当前曲目: JQ8X00_Device.Command(Play);

```

(2) void Command_Data(UartCommandData Command, uint8_t DATA);

指令+数据通信方式

UartCommandData 为枚举类型, Command 可供选择的参数如下, DATA 为配合参数的数据。

```

typedef enum
{
    AppointTrack = 0x07, /*指定曲目播放*/
    SetCycleCount = 0x19, /*设置循环次数*/
    SetEQ = 0x1A, /*EQ 设置*/
    SelectTrackNoPlay = 0x19, /*选曲不播放*/
    GoToDiskSign = 0x0B, /*切换指定盘符*/
    SetVolume = 0x13, /*音量设置*/
    SetLoopMode = 0x18, /*设置循环模式*/
    SetChannel = 0x1D, /*设置通道*/
    AppointTimeBack = 0x22, /*指定时间快退*/
    AppointTimeFast = 0x23, /*指定时间快退*/
} UartCommandData; //包含多个数据的指令

```

例如, 播放曲目 1: JQ8X00_Device.Command_Data(AppointTrack,1);特别注意, 指

定曲目播放和文件名无关，先放入的音频为曲目 1，依次类推。

(3) void AppointPlay(JQ8X00_Symbol MODE, char *DATA);

存储模块根目录下指定音频名播放

JQ8X00_Symbol 为枚举类型，MODE 可供选择的参数如下。*DATA 为指针数据，为根目录下的音频名。指定音频播放采用字符匹配形式，播放指令中包含音频的部分名字即可选中播放，例如播放 1234.MP3，输入音频名"12"即可播放根目录下 1234.MP3 音频文件。特别注意，音频名中的字母均要大写，例如：song.MP3，音频名需要改为 SONG.MP3

typedef enum

```
{  
    JQ8X00_USB = 0x00,    /*UPANND*/  
    JQ8X00_SD = 0x01,     /*SD*/  
    JQ8X00_FLASH = 0x02, /*FLASH*/  
} JQ8X00_Symbol;          //系统盘符
```

例如，播放 FLASH 根目录下 00001.mp3 音频：

```
JQ8X00_Device.AppointPlay(JQ8X00_FLASH,"00001");
```

(4) void MakeUpPlay(unsigned char *DATA, unsigned char Len);

组合播放指令

ZH 文件内的音频按照指定顺序播放。注意，ZH 文件夹内音频需要两位数字命名，如 01.MP3。

例如，依次播放 ZH 文件夹下的 02 01 音频：

```
uint8_t MakeUp_buffer[2] = {2, 1};          //组合播放内容  
JQ8X00_Device.MakeUpPlay(MakeUp_buffer, 2);
```