

QuecPython TTS 应用指导

LTE Standard 模块系列

版本：1.0.0

日期：2020-11-09

状态：临时文件



上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司
上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233
电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：
<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：support@quectel.com。

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。因未能遵守有关操作或设计规范而造成的损害，上海移远通信技术股份有限公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

免责声明

上海移远通信技术股份有限公司尽力确保开发中功能的完整性、准确性、及时性或效用，但不排除上述功能错误或遗漏的可能。除非其他有效协议另有规定，否则上海移远通信技术股份有限公司对开发中功能的使用不做任何暗示或明示的保证。在适用法律允许的最大范围内，上海移远通信技术股份有限公司不对任何因使用开发中功能而遭受的损失或损害承担责任，无论此类损失或损害是否可以预见。

保密义务

除非上海移远通信技术股份有限公司特别授权，否则我司所提供文档和信息的接收方须对接收的文档和信息保密，不得将其用于除本项目的实施与开展以外的任何其他目的。未经上海移远通信技术股份有限公司书面同意，不得获取、使用或向第三方泄露我司所提供的文档和信息。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，上海移远通信技术股份有限公司有权追究法律责任。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2020，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2020.

文档历史

修订记录

版本	日期	作者	变更表述
-	2020-11-09	Rivern	文档创建
1.0.0	2020-11-09	Rivern	临时版本

目录

文档历史	2
目录	3
1 引言	4
2 TTS 概述	5
3 TTS API 详解	6
3.1. import audio tts = audio.TTS.....	6
3.2. tts.play.....	6
3.3. tts.setSpeed	7
3.4. tts.setVolume.....	8
3.5. tts.getSpeed	8
3.6. tts.getVolume	8
3.7. tts.getState	9
3.8. tts.stop.....	9
3.9. tts.close	10
4 TTS 功能实现	11
4.1. 命令行执行	11
4.2. 执行 py 文件	13
5 附录	14

1 引言

本文档主要 EC100Y-CN 为例介绍如何使用 TTS 模块功能，帮助用户了解和使用 TTS 模块文本转语音功能。

本文档适用以下移远通信模块：

- EC100Y-CN
- EC600S-CN

2 TTS 概述

TTS 是 Text To Speech 的缩写，即“从文本到语音”，是人机对话的一部分，让机器能够说话。它是同时运用语言学和心理学的杰出之作，在内置芯片的支持之下，通过神经网络的设计，把文字智能地转化为自然语音流。TTS 技术对文本文件进行实时转换，转换时间之短可以秒计算。在其特有智能语音控制器作用下，文本输出的语音音律流畅，使得听者在听取信息时感觉自然，毫无机器语音输出的冷漠与生涩感。在使用开发板的 TTS 模块功能之前我们需要先了解模块的 Audio 接口，如下图(以 EC100Y-CN 模块为例)：

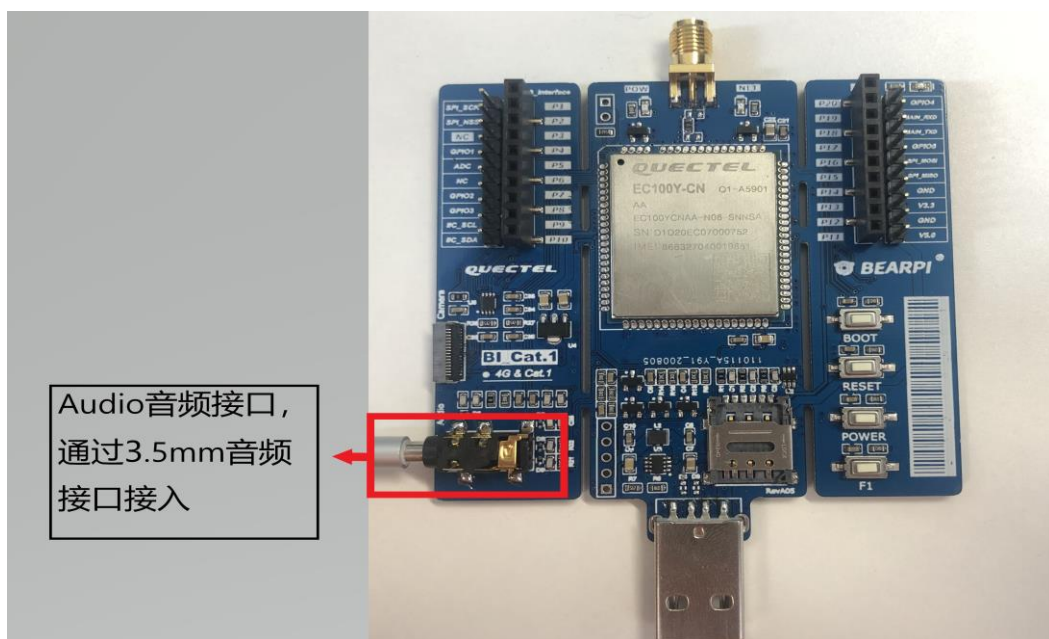


图 1: EC100Y-CN 模块 Audio 接口

连接 3.5mm 音频接口后按照《Quectel-QuecPython-Cat1 开发板使用说明》进行驱动下载和固件安装。

3 TTS API 详解

本章节介绍 TTS 相关 API。

3.1. import audio tts = audio.TTS

该函数用于导入音频（audio）库并创建 TTS 对象。

- 函数原型

```
import audio tts = audio.TTS(device)
```

- 参数

device:

表示设备类型，取值如下：

- 0 话筒
- 1 耳机
- 2 喇叭

- 返回值

无。

3.2. tts.play

该函数用于播放语音。

- 函数原型

```
tts.play(priority, breakin, mode, str)
```

- 参数

priority:

整型。播放优先级，支持优先级 0~4，数值越大优先级越高。

breakin:

整型。打断模式，0 表示不允许被打断，1 表示允许被打断。

mode:

整型。编码模式：

- 1 UNICODE16 (Size end conversion)
- 2 UTF-8
- 3 UNICODE16 (Don't convert)

str:

字符串。待播放字符串。

- 返回值

- 0 播放成功
- 1 播放失败
- 1 无法立即播放，加入播放队列
- 2 无法立即播放，且该请求的优先级组队列任务已达上限，无法加入播放队列

备注

支持优先级 0~4，数字越大优先级越高，每个优先级组可同时最多加入 10 个播放任务，播放队列策略见《Quectel QuecPython 类库 API 说明》。

3.3. tts.setSpeed

该函数用于设置播放速度。

- 函数原型

```
tts.setSpeed(speed)
```

- 参数

speed:

整型。播放速度，范围：0~9。值越大，速度越快。

- 返回值

设置成功返回当前播放速度，失败返回整型-1。

3.4. tts.setVolume

该函数用于设置播放音量。

- 函数原型

```
tts.setVolume(vol)
```

- 参数

vol:

整型。播放音量。范围：0~9，0 表示静音。

- 返回值

设置成功返回当前音量，失败返回整型-1。

3.5. tts.getSpeed

该函数用于获取当前播放速度。

- 函数原型

```
tts.getSpeed()
```

- 参数

无。

- 返回值

成功返回当前播放速度，失败返回整型-1。

3.6. tts.getVolume

该函数用于获取播放音量。

- 函数原型

```
tts.getVolume()
```

- 参数

无。

- 返回值

成功返回当前播放音量，失败返回整型-1。

3.7. tts.getState

该函数用于获取当前播放状态。

- 函数原型

```
tts.getState()
```

- 参数

无。

- 返回值

0 当前无 TTS 播放
1 当前有 TTS 正在播放

3.8. tts.stop

该函数用于暂停播放。

- 函数原型

```
tts.stop()
```

- 参数

无。

- 返回值

0 暂停播放成功

-1 暂停播放失败

3.9. tts.close

该函数用于关闭 TTS 功能。

- 函数原型

```
tts.close()
```

- 参数

无。

- 返回值

0 关闭 TTS 成功

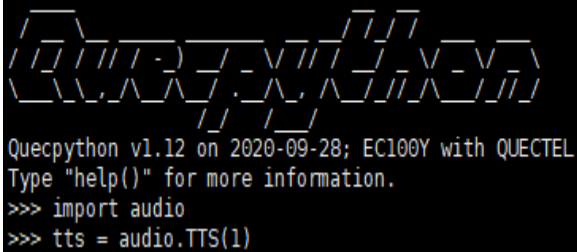
-1 关闭 TTS 失败

4 TTS 功能实现

4.1. 命令行执行

1. 通过 Xshell 连接开发板主串口后按以下步骤执行，通过以下函数导入音频（audio）库并创建 TTS 对象：

```
import audio tts = audio.TTS(device)
```



```
Quecpython v1.12 on 2020-09-28; EC100Y with QUECTEL
Type "help()" for more information.
>>> import audio
>>> tts = audio.TTS(1)
```

2. 使用以下函数播放语音：

```
tts.play(priority, breakin, mode, str)
```

```
>>> tts.play(1,0,2,"11111111")
0
>>> tts.play(1,0,2,"1111111111111111")
0
>>> tts.play(2,0,2,"1111111111111111")
1
```

```
>>> tts.stop()
>>> for i in range(12):
...     tts.play(0,0,2,'222222222')
...
0
1
1
1
1
1
1
1
1
1
1
1
1
-2
>>> █
```

3. 此时可在耳机中听到播放文本内容的语音，可通过 `tts.setSpeed(speed)`和 `tts.setVolume(vol)`来设置播放速度和播放音量。

```
>>> tts.setVolume(7)
0
>>> tts.setSpeed(8)
0
>>> █
```

4. 通过 `tts.getSpeed()`和 `tts.getVolume()`来获取当前播放速度和音量。另外，可通过 `tts.getState()`来获取当前播放状态。

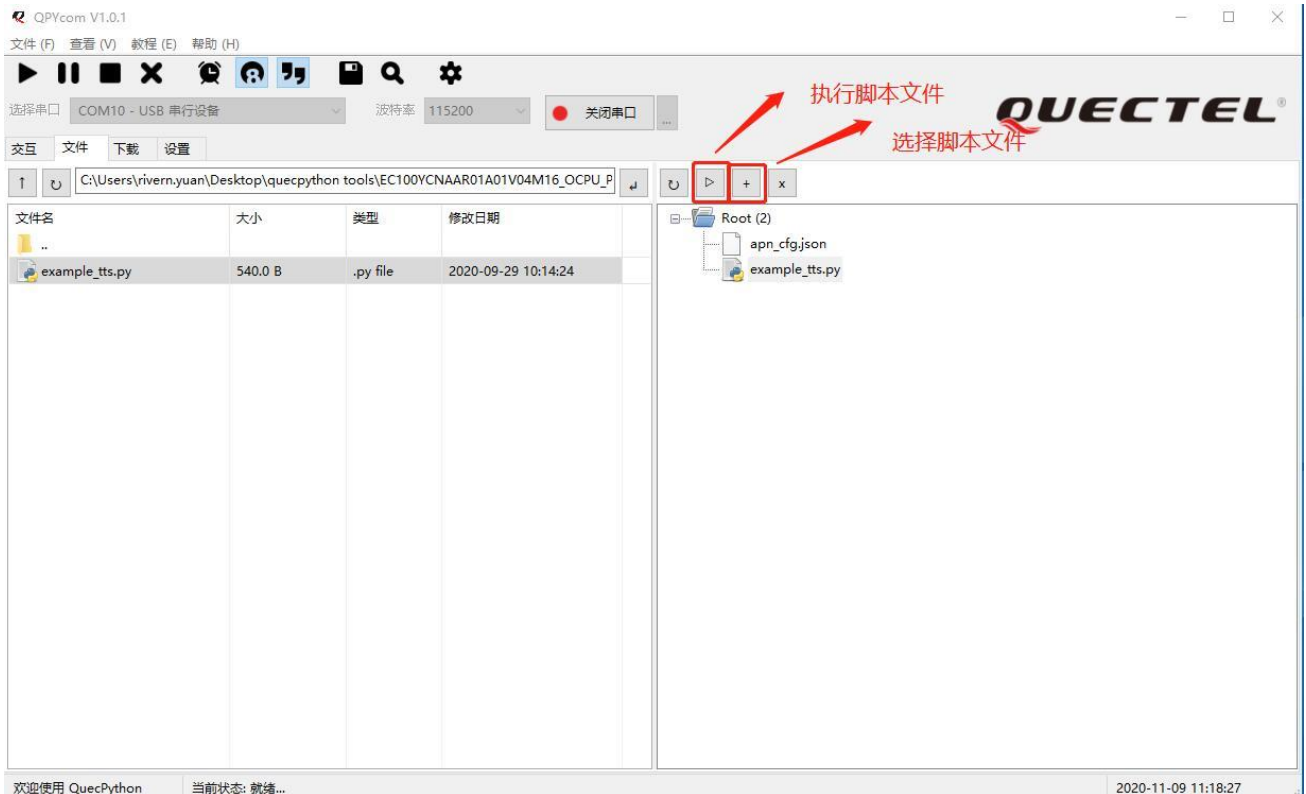
```
>>> tts.getSpeed()
8
>>> tts.getVolume()
7
>>> tts.getState()
0
>>> █
```

5. 播放过程中可通过 `tts.stop()`来暂停播放。播放完成后可通过 `tts.close()`关闭 TTS 功能。

```
>>> help(tts)
object tts is of type TTS
  play -- <function>
  stop -- <function>
  close -- <function>
  setVolume -- <function>
  getVolume -- <function>
  setSpeed -- <function>
  getSpeed -- <function>
  getState -- <function>
>>> tts.stop()
>>> tts.close()
0
>>> █
```

4.2. 执行 py 文件

1. 首先在提供的 SDK 工具包中进入 demo 目录，找到 TTS 文件夹，通过 QPYcom 工具将 TTS 目录下的 example_tts.py 脚本文件发送到模块中，脚本文件发送和执行具体步骤，参见《QPYcom 工具使用说明》。



2. 在 Xshell 中，连接模块主串口，进入交互界面，通过 `uos.listdir()`方法确认上一步骤发送的脚本文件是否在当前目录下，然后执行如下步骤：

```
>>> import uos
>>> uos.listdir()
['apn_cfg.json', 'example_tts.py', 'example_socket.py', 'example_thread.py']
>>> import example
>>> example.exec('example_tts.py')
>>> []
```

通过 `import example` 导入 `example` 模块，该模块提供了 `exec()`方法用来执行 python 脚本程序，通过 `example.exec('example.tts.py')`来执行 `example.tts.py` 脚本，执行完成后即可在耳机中听到播放脚本中文本内容的语音。

5 附录

表 1：术语缩写

术语	英文全称	中文全称
API	Application Programming Interface	应用程序编程接口
LTE	Long Term Evolution	长期演进
SDK	Software Development Kit	软件开发工具包
TTS	Text To Speech	从文本到语音
UNICODE	Unicode	统一码
UTF	Universal Character Set/Unicode Transformation Format	针对 Unicode 的可变长度字符编码