

QuecPython Record 用户指导

版本：1.0.0

日期：2021-01-07

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 2021，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2021.

文档历史

修订记录

版本	日期	作者	变更表述
-	2021-01-06	Felix YE	文档创建
1.0.0	2021-01-07	Felix YE	临时版本

目录

文档历史	2
1 引言	4
2 用户接口	5
2.1. 创建 RECORD 对象	5
2.2. 开始录音	5
2.3. 停止录音	6
2.4. 读取录音文件的路径	6
2.5. 读取录音数据	7
2.6. 读取录音文件大小	7
2.7. 删除录音文件	7
2.8. 判断录音文件是否存在	8
2.9. 判断是否正在录音	8
3 附件	9
3.1. record 流程脚本命令	9
3.2. 脚本执行结果	9

1 引言

本文档基于 QuecPython 介绍如何通过 *record* 模块实现录音功能，包括参数说明，如何创建 *record* 对象，录音控制，以及录音文件信息的获取和播放等。

适用模块：

- EC100Y-CN（本文以该模块为例进行介绍）
- EC600S-CN

2 用户接口

2.1. 创建 RECORD 对象

1. 导入 audio 模块
创建 RECORD 对象

示例代码参考以下代码清单：

```
import audio
record = audio.Record(file_name,callback)
```

2. 参数列表

参数	参数类型	参数说明
file_name	str	录音文件名
callback	api	录音结束的回调

3. 返回值
-1: err
对象: 成功

4. 示例

```
def record_callback(para):
    print("file_name:",para[0])    //返回文件路径
    print("audio_len:",para[1])    //返回录音长度
    print("audio_state:",para[2])  //返回录音状态  -1: error
                                         3: 成功
```

```
import audio
record_test = audio.Record("record_test.wav",record_callback)
```

2.2. 开始录音

1. 调用方法

```
record.start(seconds)
```

2. 参数列表

参数	参数类型	参数说明
seconds	int	需要录制时长，单位：秒

3. 返回值
 - 0: 成功
 - 1: 文件覆盖失败
 - 2: 文件打开失败
 - 3: 文件正在使用
 - 4: 通道设置错误（只能设置 0 或 1）
 - 5: 定时器资源申请失败
4. 示例:


```
record_test.start(40)
```

2.3. 停止录音

1. 调用方法


```
record.stop()
```
2. 参数列表

无
3. 返回值

无
4. 示例:


```
record_test.stop()
```

2.4. 读取录音文件的路径

1. 调用方法


```
record.getFilePath()
```
2. 参数列表

无
3. 返回值

String: 录音文件的路径
4. 示例:


```
record_test.getFilePath()
```

2.5. 读取录音数据

1. 调用方法

```
record.getData(offset, size)
```

2. 参数列表

参数	参数类型	参数说明
offset	int	读取数据的偏移量
size	Int	读取大小：需小于 10K

3. 返回值

- 1: 读取数据错误
 - 2: 文件打开失败
 - 3: 偏移量设置错误
 - 4: 文件正在使用
 - 5: 设置超出文件大小 (offset+size > file_size)
 - 6: 读取 size 大于 10K
 - 7: 内存不足 10K
- bytes:返回数据

4. 示例:

```
record_test.getData(0, 44) //读取文件头
```

2.6. 读取录音文件大小

1. 调用方法

```
record.getSize()
```

2. 参数列表

无

3. 返回值

- 1: 获取文件 size 失败
 - 2: 文件打开失败
 - 3: 文件正在使用
- size: 文件大小 //此值会比返回 callback 返回值大 44byte (44byte 为文件头)

4. 示例:

```
record_test.getSize()
```

2.7. 删除录音文件

1. 调用方法

```
record.Delete()
```


2. 参数列表
无
3. 返回值
0: 成功
-1: 文件不存在
-2: 文件正在使用
4. 示例:
`record_test.Delete()`

2.8. 判断录音文件是否存在

1. 调用方法
`record.exists()`
2. 参数列表
无
3. 返回值
`true:` 文件存在
`false:` 文件不存在
4. 示例:
`record_test.exists()`

2.9. 判断是否正在录音

1. 调用方法
`record.isBusy()`
2. 参数列表
无
3. 返回值
0: idle
1: busy
4. 示例:
`record_test.isBusy()`

3 附件

3.1. record 流程脚本命令

record 流程的脚本命令汇总如下：

```
def record_callback(para):  
    print("file_name:", para[0])  
    print("audio_len:", para[1])  
    print("audio_state:", para[2])  
  
import audio  
record_test = audio.Record("record_test.wav", record_callback)  
  
record_test.isBusy()  
  
record_test.start(40)  
  
record_test.isBusy()  
record_test.stop()  
  
record_test.getFilePath()  
  
record_test.getData(0, 44)  
  
record_test.getSize()  
  
record_test.Delete()  
  
record_test.exists()  
  
record_test.isBusy()
```

3.2. 脚本执行结果

脚本执行结果分别下图所示：

```

>>>
>>> def record_callback(para):
...     print("file_name:",para[0])
...     print("audio_len:",para[1])
...     print("audio_state:",para[2])
...
>>> import audio
>>> record_test = audio.Record("record_test.wav",record_callback)
>>>
>>> record_test.isBusy()
0
>>> record_test.start(40)
0
>>> record_test.isBusy()
1
>>> record_test.stop()
>>> file_name: U:/record_test.wav
audio_len: 216000
audio_state: 3
record_test.getFilePath()
'U:/record_test.wav'
>>> record_test.getData(0, 44)
b'RIFF\x00\x00\x00WAVEfmt \x10\x00\x00\x01\x00\x01\x00@\x1f\x00\x00\x80>\x0
0\x00\x02\x00\x10\x00data\x00\x00\x00\x00'
>>> record_test.getSize()
216044
>>> record_test.exists()
True
>>> record_test.Delete()
0
>>> record_test.exists()
False
>>> record_test.isBusy()
0
>>>

```