

QuecPython 基础操作说明

LTE Standard 模块系列

版本：1.0.0

日期：2020-11-13

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
-	2020-05-06	Baron	文档创建
1.0.0	2020-11-13	Jayceon/Kingka	临时版本

目录

文档历史	2
图片索引	4
1 引言	5
2 系统启动	6
3 文件系统	8
3.1. 脚本下载	8
3.2. 查看文件	9
3.3. 命令交互	10
4 操作脚本	11
5 字节码编译	12
6 附录 A 术语缩写	13

图片索引

图 1: 交互式解释器	6
图 2: 脚本下载界面	9
图 3: 本地与模块进行 Python 文件的上传、查看、添加、删除操作	9
图 4: 交互界面说明	10
图 5: 执行脚本界面	11

1 引言

本文主要介绍 QuecPython 基础操作，包括文件系统以及指令执行。

在 QuecPython 中，使用主串口作为指令和数据接收通道，所有操作都通过主串口完成。

适用模块：

- EC100Y-CN（本文以该模块为例进行介绍）
- EC600S-CN

2 系统启动

QuecPython 启动后，将在主串口启动交互式解释器，类似于 Linux Shell。通过该交互式解释器，用户可实时地执行命令，并查看返回的结果。

```

Quecpython v1.12 on 2020-05-06; EC100Y with QUECTEL
Type "help()" for more information.
>>> help()
Welcome to QuecPython on the EC100Y!

For access to the hardware use the 'machine' module:

import machine
m = machine.lcd()
m.lcd_init()
m.lcd_brightness(5)
m.lcd_show()

Basic socket operation:

import usocket
sock = usocket.socket(usocket.AF_INET, usocket.SOCK_STREAM)
sockaddr = usocket.getaddrinfo('ifconfig.me', '80')[0][-1]
sock.connect(sockaddr)
sock.send('GET /ip HTTP/1.1\r\nHost: ifconfig.me\r\nConnection: keep-alive\r\n\r\n')
sock.recv(256)

Control commands:
CTRL-A      -- on a blank line, enter raw REPL mode
CTRL-B      -- on a blank line, enter normal REPL mode
CTRL-C      -- interrupt a running program
CTRL-D      -- on a blank line, do a soft reset of the board
CTRL-E      -- on a blank line, enter paste mode

For further help on a specific object, type help(obj)
For a list of available modules, type help('modules')

```

图 1：交互式解释器

其中：

- 执行 **help(obj)** 指令，查看帮助；
- 执行 **dir(obj)** 指令，查看模块提供的详细方法；
- 执行 **help('modules')** 指令，查看当前支持的类库。

系统启动时，除了进行硬件资源初始化外，还会进行分区挂载，执行初始化脚本操作。启动脚本主要包括两个：

- `__boot.py`: 资源初始化等，如启动时挂载分区，该脚本被冻结在出厂固件中；
- `main.py`: 用户初始化脚本，系统初始化完成后执行。

3 文件系统

在 QuecPython 中，划分了 5 MB 的空间作为用户分区，用户可以将一些配置、脚本等文件存储在该分区中。在系统启动时，会自动挂载该分区，该分区挂载在 ‘/’ 目录。

在 QuecPython 中，提供了对文件系统访问的类库 *uos*，可用于操作和访问文件系统。

以下代码示例为如何在当前目录下创建文件、写入内容及读取内容。

```
import uos

# create a file
f=open('test.txt','w')
f.write('hello quecpython!\n')
f.write('123456789abcdefg!\n')
f.close()
# read a file
f=open('test.txt','r')
print(f.readline())
print(f.readline())
f.close()
```

为方便操作，可使用 *QPYcom.exe* 工具进行常规的文件系统操作。

3.1. 脚本下载

- 步骤 1: 解压 SDK 压缩包内 *tools* 目录下的 *QPYcom.zip*，获取 *QPYcom.exe*，并双击运行；
- 步骤 2: 进入“下载”Tab，点击“创建”按钮，根据需求创建用户项目；
- 步骤 3: 点击“+”按钮，选择需要下载到模块的脚本；
- 步骤 4: 点击界面右下方倒三角按钮，切换到“下载脚本”模式；
- 步骤 5: 点击“Download FW”按钮，下载脚本；
- 步骤 6: 进度条显示为“100%”时，表示下载完成，可进入“文件”Tab 查看模块内文件详情。



图 2：脚本下载界面

3.2. 查看文件

运行 QPYcom.exe 工具，点击“查看”-->“文件浏览”，可实现本地与模块进行 Python 文件的上传、查看、添加、删除操作，操作界面及按钮如下图所示。

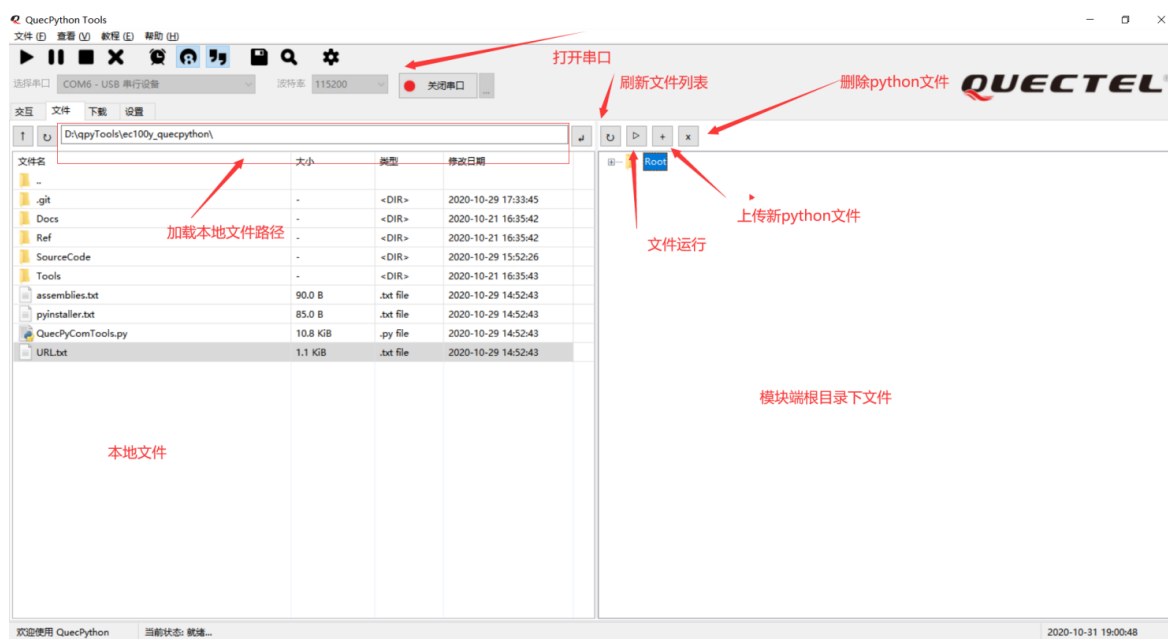


图 3：本地与模块进行 Python 文件的上传、查看、添加、删除操作

3.3. 命令交互

运行 QPYcom.exe 工具，点击“查看”-->“交互命令行”，进入交互主界面。在交互界面可以通过交互窗口与模块进行手动输入交互，交互主界面说明如图所示。

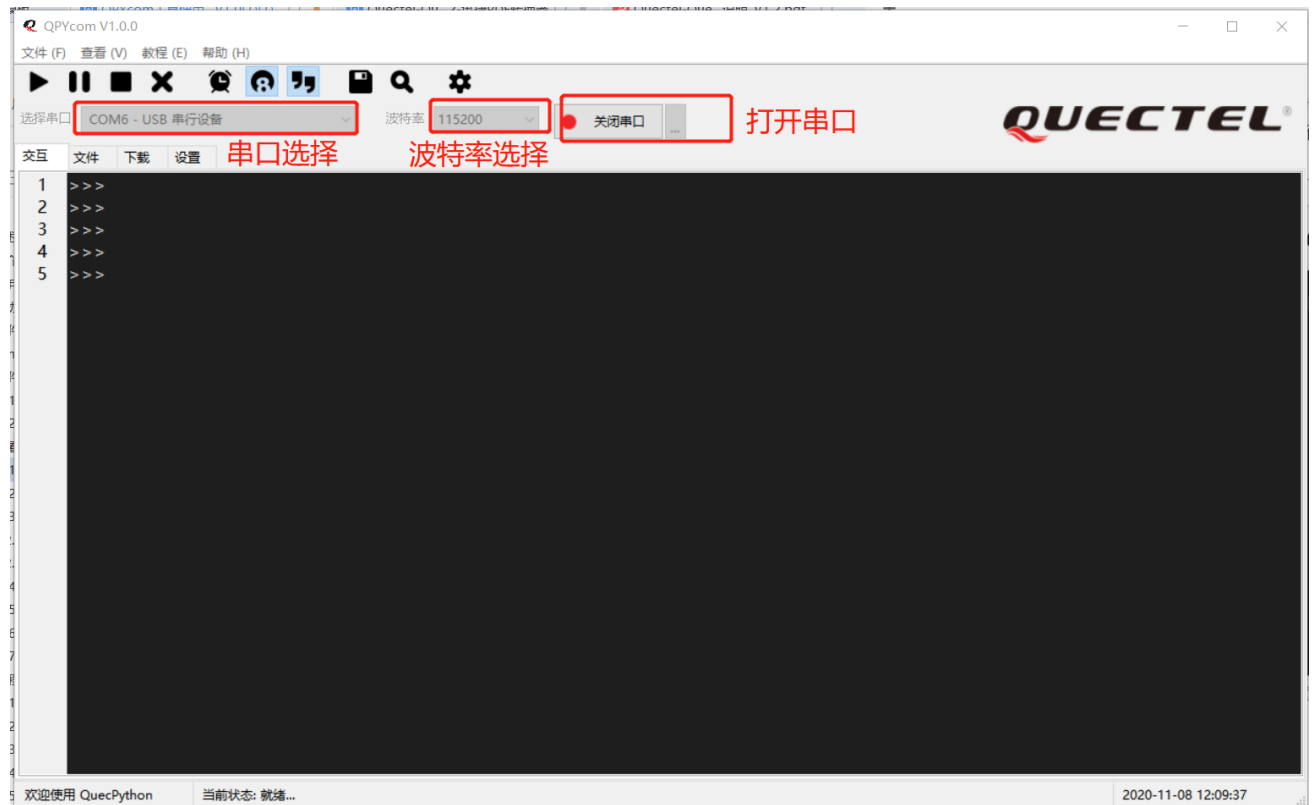


图 4：交互界面说明

4 执行脚本

为方便本地测试，无需将脚本文件上传到模块执行，**QPYcom.exe** 工具提供了在模块端直接执行本地脚本的方法。

步骤 1：解压 SDK 压缩包内 *tools* 目录下的 *QPYcom.zip*，获取 *QPYcom.exe*，并双击运行；

步骤 2：进入“文件”Tab；

步骤 3：点击下图红框所示按钮，执行脚本文件。

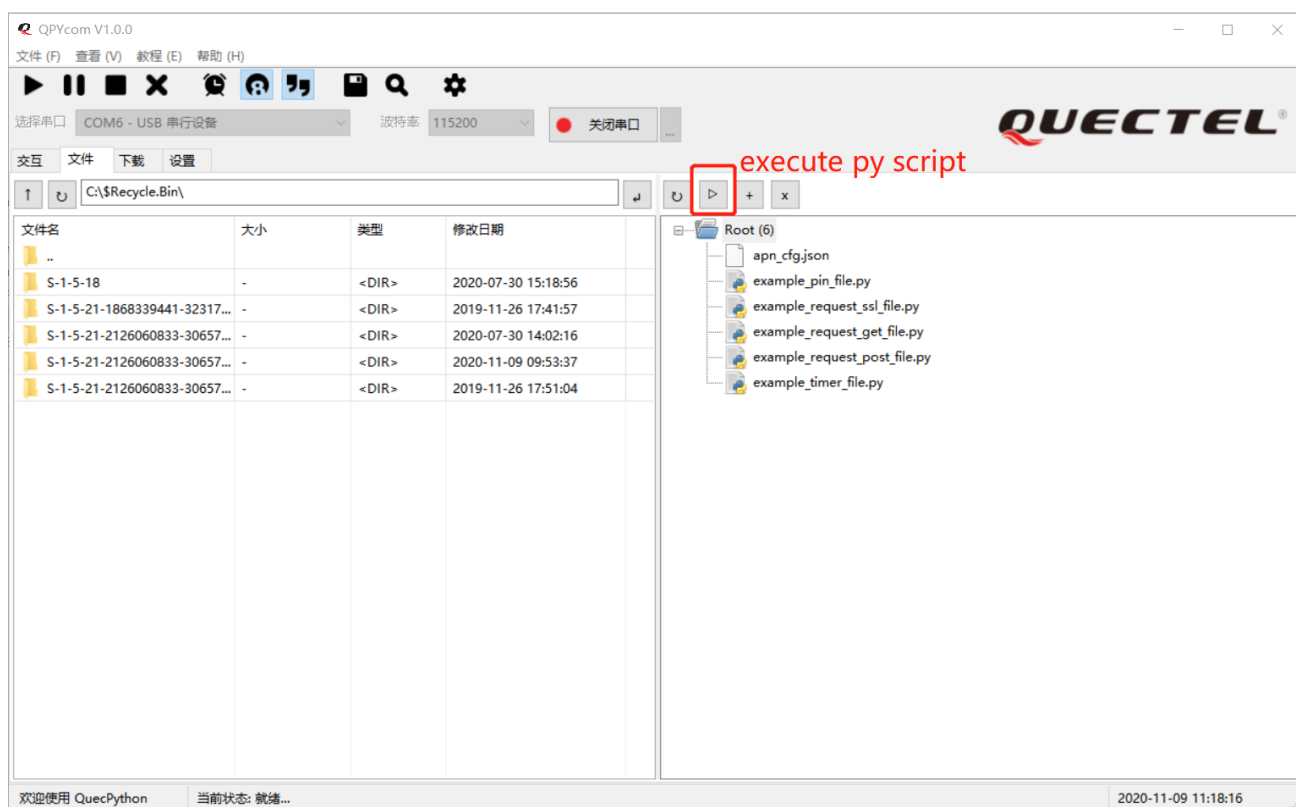


图 5：执行脚本界面

5 字节码编译

为了提高代码执行速度以及客户代码安全，移远通信提供了 `mpy-cross` 工具将用户 Python 脚本编译为字节码。字节码可固化在固件中，也可以存放在文件系统中供脚本使用。详情请参考《Quectel_QuecPython_mpy-cross 用户指导》

Python 脚本编译为字节码的命令为：

```
mpy-cross.exe -o test.mpy -s test.py -march=armv7m test.py
```

6 附录 A 术语缩写

表 1：术语缩写

缩写	英文全称	中文全称
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器