

QuecPython mpy-cross 用户指导

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	Kingka/ Kenney	文档创建
1.0.0	2020-11-09	Kingka/ Kenney	临时版本

目录

文档历史	2
目录	3
图片索引	4
1 引言	5
2 工具使用	6
2.1. 工具介绍	6
2.1.1. 文件说明	6
2.1.2. 参数说明	6
2.2. 工具使用	7

图片索引

图 1: mpy-cross 参数使用说明 6

图 2: 开发板接入电脑..... 7

图 3: 命令行进入工具目录..... 7

图 4: mpy 文件..... 8

图 5: test.py 运行结果 8

1 引言

为提升用户 Python 源文件的安全性及文件执行效率，移远通信模块使用 mpy-cross 工具对 Python 源代码文件（.py）进行加密，加密后生成二进制字节码文件（.mpy）。

本文以 EC100Y-CN 模块为例介绍如何使用 mpy-cross 工具来完成 Python 源代码文件加密。

本文档适用如下移远通信模块：

- EC100Y-CN
- EC600S-CN

2 工具使用

2.1. 工具介绍

在 Python 中，可将.py 文件编译为.pyc 文件。编译后的.pyc 文件是二进制格式，可加快加载速度，更重要的是可以保护原始代码。micropython 使用 mpy-cross 工具完成.py 文件编译并加密，编译后的文件为.mpy。

2.1.1. 文件说明

- .py 文件

Python 源代码文件

- .pyc 文件

二进制文件，Python 源代码文件经过编译后生成的字节码文件。.pyc 文件加载速度有所提高，且 pyc 是一种跨平台的字节码，由 Python 虚拟机来执行。

- .mpy 文件

Micropython 提供 mpy-cross 工具，用于将 Python 源代码文件编译成.mpy 文件。该文件和.pyc 文件均为二进制字节码文件。

2.1.2. 参数说明

```
C:\Users\User\Desktop\DOCS\EC100Y\Quecpython\QuecPython-SDK-EC100Y-V0.4\tools\QuecPyComTools\QuecPyComTools>mpy-cross-amd64.exe -h
usage: mpy-cross-amd64.exe [<opts>] [-X <implopt>] <input filename>
Options:
  --version : show version information
  -o : output file for compiled bytecode (defaults to input with .mpy extension)
  -s : source filename to embed in the compiled bytecode (defaults to input file)
  -v : verbose (trace various operations); can be multiple
  -O[N] : apply bytecode optimizations of level N

Target specific options:
  -msmall-int-bits=number : set the maximum bits used to encode a small-int
  -mno-unicode : don't support unicode in compiled strings
  -mcache-lookup-bc : cache map lookups in the bytecode
  -march=<arch> : set architecture for native emitter; x86, x64, armv6, armv7m, armv7em, armv7emsp, armv7emdp, xtensa, xtensawin

Implementation specific options:
  -emit={bytecode,native,viper} -- set the default code emitter
  -heapsize=<n> -- set the heap size for the GC (default 2097152)
```

图 1: mpy-cross 参数使用说明

备注

可访问如下链接获取更多有关工具 mpy-cross 的说明：

<https://pypi.org/project/mpy-cross/1.9.3/>

https://makeblock-micropython-api.readthedocs.io/zh/latest/novapi/tutorial/precompiled_to_mpy.html

2.2. 工具使用

1. 连接 EC100Y-CN 开发板至电脑，如下图所示。接入后操作方法详见《Quectel_QuecPython_基础操作说明》。

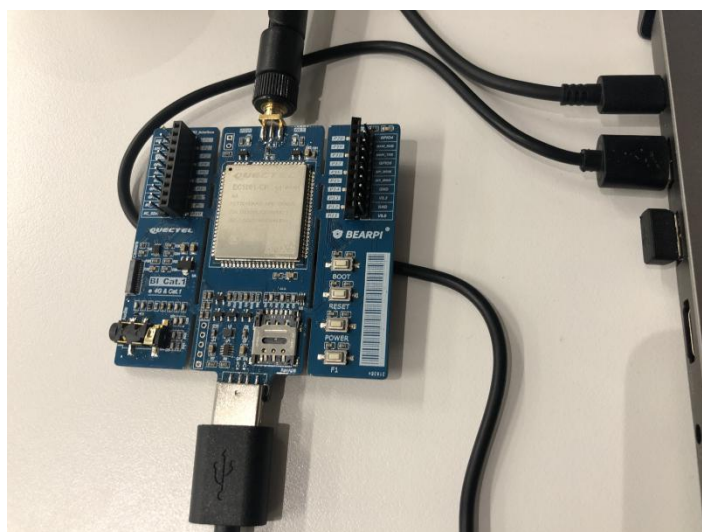


图 2：开发板接入电脑

2. 使用 mpy-cross-amd64.exe 工具。在工具同目录下编写 `usertest.py` 文件，文件内编写测试函数，如下所示：

```
def test_mpy():
    print("hello this is mpy file")
```

3. 打开 Windows 下 cmd 命令行，进入 mpy-cross-amd64.exe 工具所在目录，如下图所示。

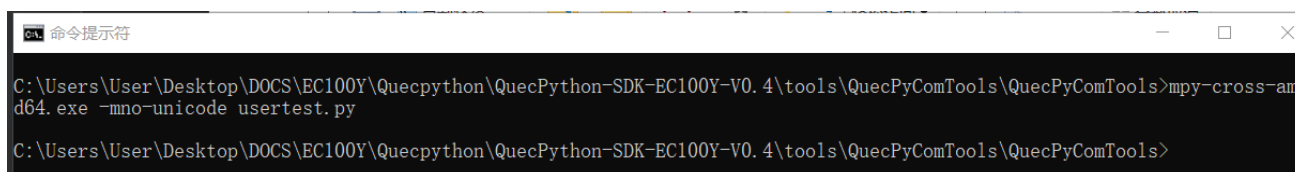


图 3：命令行进入工具目录

- 随后执行如下命令及参数生成.mpy 文件:

```
mpy-cross-amd64.exe -mno-unicode usertest.py
```

生成的.mpy 如下图所示




 mpy-cross-amd64.exe	2019/12/21 17:10	应用程序	357 KB
 usertest.mpy	2020/10/27 16:43	MPY 文件	1 KB
 usertest.py	2020/10/27 16:37	Python File	1 KB


图 4: mpy 文件

- 在 *test.py* 文件中使用 **import** 导入 *usertest* 模块，直接调用 *usertest* 文件中的方法，如下所示:

```
import usertest

usertest.test_mpy()
```

- 把 *test.py* 文件和 *usertest.mpy* 文件分别上传到移远通信 EC100Y-CN 开发板内，上传方法详见《Quectel_QuecPython_基础操作说明》。
- 在开发板中运行 *test.py* 文件，可看到经过 mpy-cross 工具加密过的 *usertest* 模块执行结果，如下图所示。



```

Quecpython v1.12 on 2020-10-26; EC100Y with QUECTEL
Type "help()" for more information.
>>>
>>> import example
>>> example.exec('test.py')
hello this is mpy file
>>>

```

图 5: test.py 运行结果