QuecPython mpy-cross User Guide

LTE Standard Module Series

Version: 1.0.0

Date: 2020-11-09 Status:

Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit: http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm Or email to support@quectel.com.

General Notes

Quectel offers the information as a service to its customers. The information provided is based upon customers' requirements. Quectel makes every effort to ensure the quality of the information it makes available. Quectel does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. All information supplied herein is subject to change without prior notice.

Disclaimer

While Quectel has made efforts to ensure that the functions and features under development are free from errors, it is possible that these functions and features could contain errors, inaccuracies and omissions. Unless otherwise provided by valid agreement, Quectel makes no warranties of any kind, implied or express, with respect to the use of features and functions under development. To the maximum extent permitted by law, Quectel excludes all liability for any loss or damage suffered in connection with the use of the functions and features under development, regardless of whether such loss or damage may have been foreseeable.

Duty of Confidentiality

The Receiving Party shall keep confidential all documentation and information provided by Quectel, except when the specific permission has been granted by Quectel. The Receiving Party shall not access or use Quectel's documentation and information for any purpose except as expressly provided herein. Furthermore, the Receiving Party shall not disclose any of the Quectel's documentation and information to any third party without the prior written consent by Quectel. For any noncompliance to the above requirements, unauthorized use, or other illegal or malicious use of the documentation and information, Quectel will reserve the right to take legal action.

Copyright

About the Document

Revision History

Version	Date	Author	Description
-	2020-11-09	Kingka/ Kenney	Creation of the document
1.0.0	2020-11-09	Kingka/ Kenney	Preliminary

Contents

Figure Index

1 Introduction

To improve the security of Python source files and the execution efficiency of file, Quectel modules encrypt Python source code files (.py) with mpy-cross tool, and then generates binary bytecode files (.mpy) after encryption.

This document takes EC100Y-CN module as an example to show how to encrypt the Python source code files with mpy-cross tool.

2 Use of mpy-cross Tool

2.1. Introduction of mpy-cross Tool

In Python, the source code file (.py) can be compiled to .pyc file in binary format. The .pyc file accelerates loading speed, and the most important is the .pyc file protects the source codes. mpy-cross is the micropython cross compiler utility, used to pre-compile python files into bytecode suitable. The generated file is .mpy file.

2.1.1. File Description

- .py file

  Python source codes file.

- .pyc file

  Bytecode file in binary format generated after the Python source codes file is compiled. The loading speed of .pyc file is quicker and is executed on Python virtue machine.

- .mpy file

  Bytecode file in binary format generated by mpy-cross tool provided by micropython.

2.1.2. Parameter Description

```
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)
```

Figure 1: Parameter Description of mpy-cross

NOTE

Please access the following URLs for details about mpy-cross:
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. Use of mpy-cross Tool

1. Connect EC100Y-CN EVB to PC as the figure below shows. Please refer to *Quectel_QuecPython_Basic_Operation_Guide* for the operations after the EVB is connected.

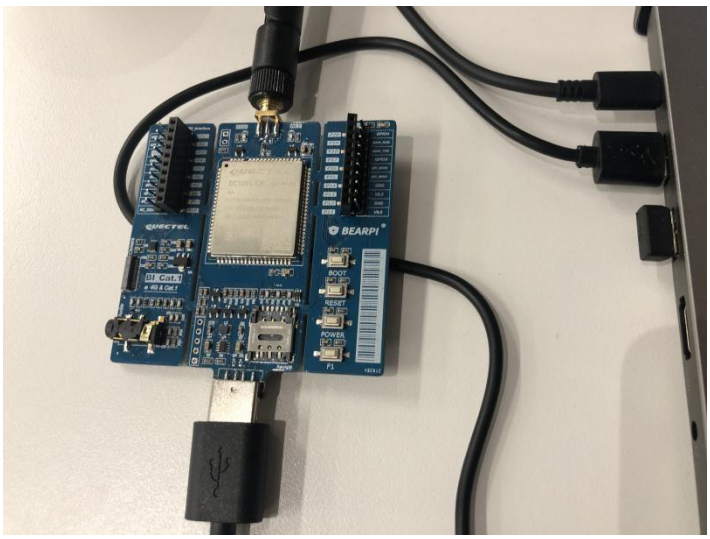

Figure 2: Connect EVB to PC

2. Open tool mpy-cross-amd64.exe and write test function in the file *usertest.py* under the same directory as the tool.

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

3. Open cmd command line in Windows system and enter the directory of tool mpy-cross-amd64.exe.

Figure 3: Enter the Directory of the Tool

4. Execute the following command to generate .mpy file.

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

The generated file is shown as below:

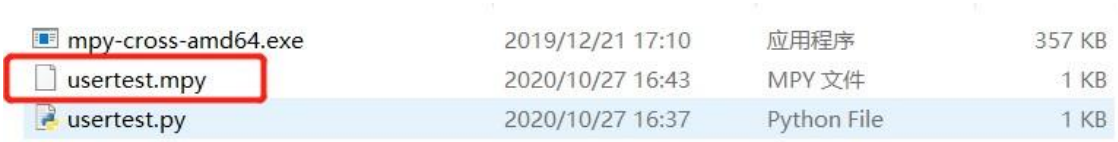

Figure 4: .mpy File

5. Import usertest module with **import** in file *test.py* and call methods of file *usertest* directly.

```
import usertest

usertest.test_mpy()
```

6. Upload the files *est.py* and *usertest.mpy* to Quectel EC100Y-CN EVB. Please refer to *Quectel_QuecPython_Basic_Operation_Guide* for uploading method.

7. Run file *test.py* in EC100Y-CN EVB. The running result of usertest module encrypted with mpy-cross is viewable.

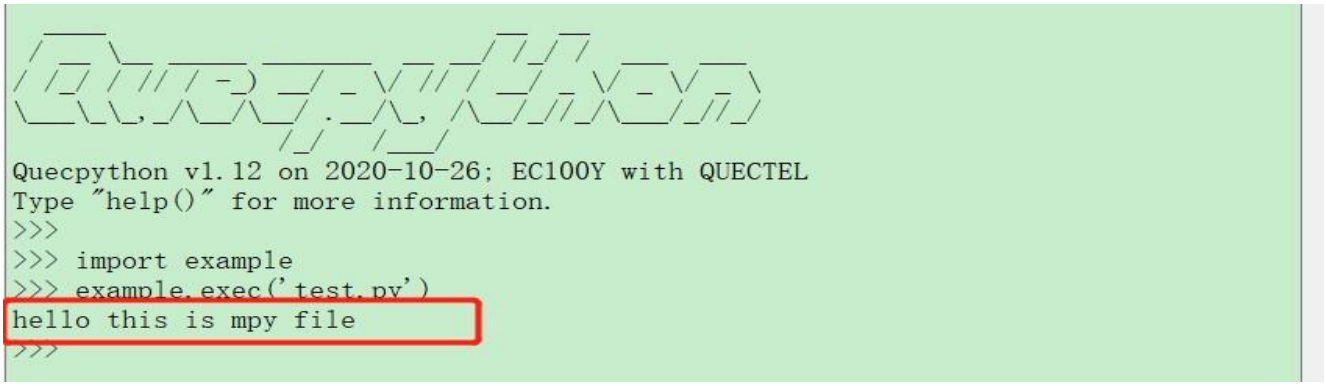

Figure 5: Running Result

3 Appendix A References

Table 1: Terms and Abbreviations

Abbreviation	Description
EVB	Evaluation Board
PC	Personal Computer