QuecPython

FILE User Guide

LTE Standard Module Series

Version: 1.0.0

Date: 2020-11-09

Status: Preliminary

www.quectel.com

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

History

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

Contents

Table Index

Figure Index

1 Introduction

This document describes how to use the QuecPython class library API to quickly develop the file read-write function on QuecPython platform.

The applicable modules:

- EC100Y-CN (This document takes this module as an example)
- EC600S-CN

2 Basic Concept of File

Files save and store data on a long-term storage device, which mainly includes hard disk, U disk, mobile hard disk, optical disk, etc.

2.1. File Storage Form

Files are mainly stored in the form of binary and text.

1. Text file, such as Python source program:
   - It can be viewed using text editing software.
   - Essentially a binary file.

2. Binary file, such as picture files, audio files, video files:
   - The saved content cannot be read directly, but is provided to other software.
   - It cannot be viewed by using text editing software.

2.2. Basic Operation

2.2.1. File Operation Type

- Open the file.
- Read and write the file.
  Read: Read the content of the file into the memory.
  Write: Write the content of the memory into the file.
- Close the file.

2.2.2. File Access Mode

Table 1: File Access Mode

Access Mode	Description
r	Open the file in read-only mode. The pointer to the file will be placed at the beginning of the file. If the file does not exist, ERROR will be occurred.

w	Open the file in write-only mode. If the file exists, it will be overwritten; if it does not exist, a new file will be created.
a	Open the file in read-write mode. If the file exists, the pointer will be placed at the end of the file; If it does not exist, a new file will be created.

NOTE

If the file pointer is moved frequently, it will affect the efficiency of file reading and writing. Generally, files are manipulated in a read-only and write-only mode during the development process.

2.2.3. File Operation Function

Table 2: File Operation Function

Function	Description	Method
Open	Open the file, and the file object will be returned.	In charge of opening file and returning the file object.
Read	Read the content of file into the memory.	Called by the file object.
Write	Write the specified content into the file.	
Close	Close the file	

3 Operate the File or Directory

Connect the QuecPython EVB to PC, and please refer to *Quectel_QuecPython_Basic_Operation_Introduction*.

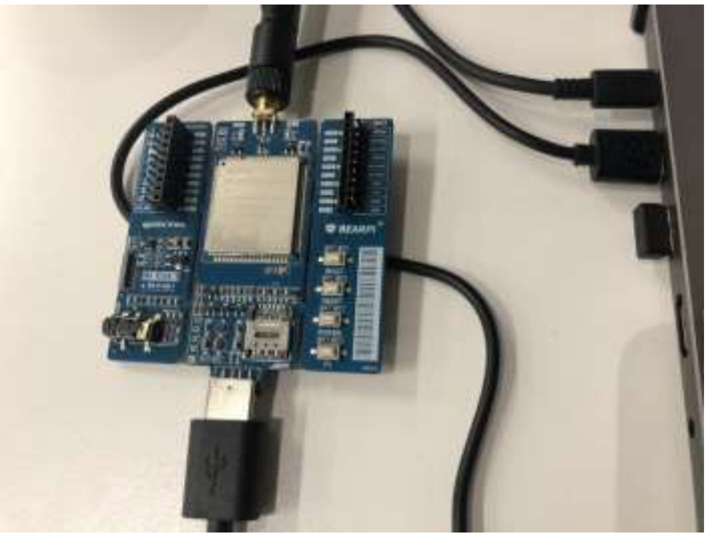


Figure 1: Connect QuecPython EVB to PC

3.1. Open the File in Read-Only Mode

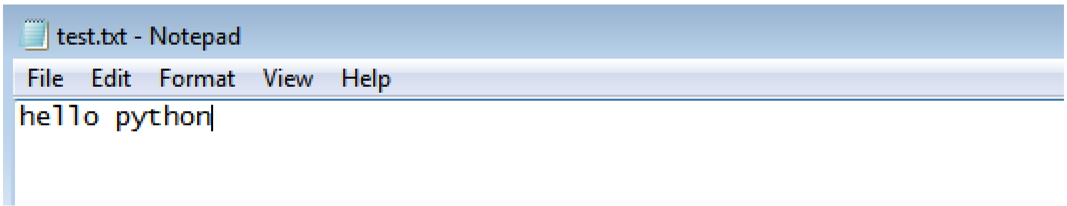
Step 1: Create the file *test.py* and *test.txt*, and import *uio* module to *test.py*, and write "hello python" into *test.txt* file.

- Import *uio* module of QuecPython to *test.py*.

```
import uio

# 1. Open test.txt file in read-only mode.
fd = uio.open("test.txt", mode='r')

# 2. Read the file content.
text = fd.read()
print(text)
```

```
# 3. Close the file
fd.close()
```

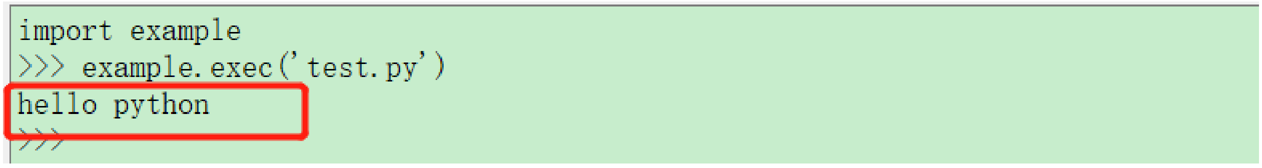
- Write "hello python" into *test.txt*.



```
hello python
```

Step 2: Upload the file *test.py* and *test.txt* respectively to QuecPython EVB, please refer to *Quectel_QuecPython_Basic_Operation_Introduction*.

Step 3: The running result for reading file.



```
import example
>>> example.exec('test.py')
hello python
>>>
```

Figure 2: The Running Result for Reading File

3.2. Open the File in Write-Only Mode

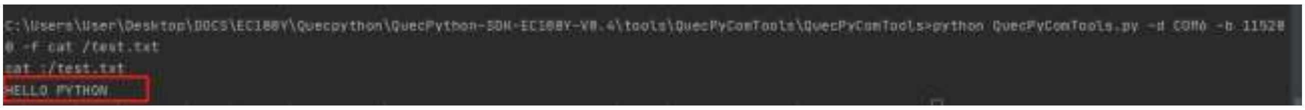
Step 1: Create the file *test.py* and *test.txt* with empty content, and import *uio* module to *test.py*, then write the following code.

```
import uio

# 1. Open test.txt file in write-only mode.
fd = uio.open("test.txt", mode='w')

# 2. Write the file into the test.txt file.
fd.write("HELLO PYTHON")

# 3.Close the file
fd.close()
```

Step 2: Upload the file *test.py* and *test.txt* respectively to QuecPython EVB, please refer to

Quectel_QuecPython_Basic_Operation_Introduction.

Step 3: The running result for writing file.

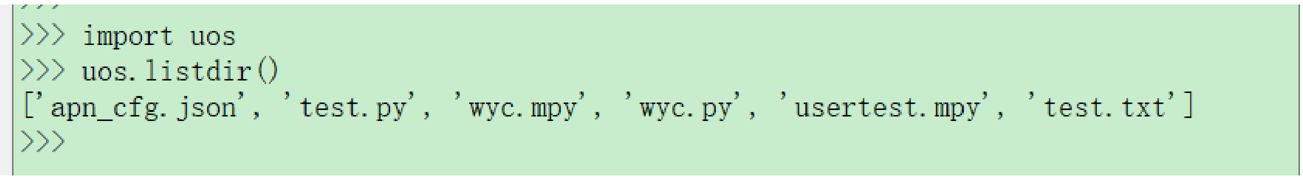


```
C:\Users\User\Desktop\DOCS\EC100Y\Quecpython\QuecPython-SDK-EC100Y-V0.4\tools\QuecPyComTools\QuecPyComTools>python QuecPyComTools.py -d COM6 -b 11520
0 -f cat /test.txt
cat :/test.txt
HELLO PYTHON
```

Figure 3: The Running Result for Writing File

3.3. Use uos Module

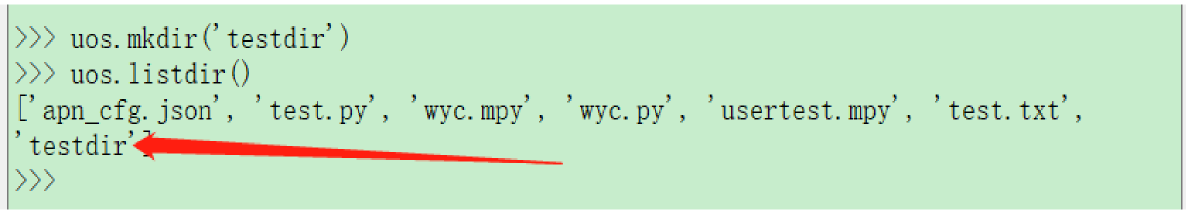
1. List the current file.



```
>>> import uos
>>> uos.listdir()
['apn_cfg.json', 'test.py', 'wyc.mpy', 'wyc.py', 'usertest.mpy', 'test.txt']
>>>
```

Figure 4: List the Current File

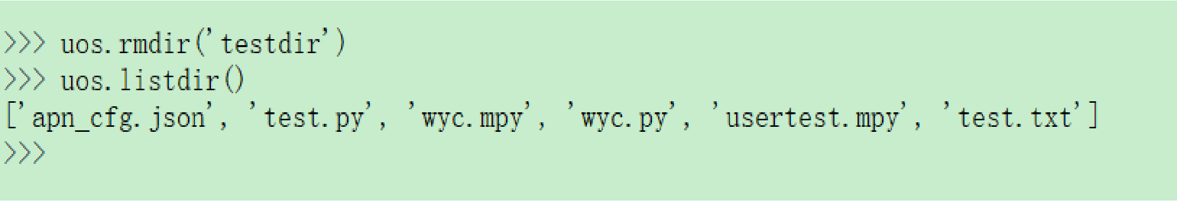
2. Create the directory



```
>>> uos.mkdir('testdir')
>>> uos.listdir()
['apn_cfg.json', 'test.py', 'wyc.mpy', 'wyc.py', 'usertest.mpy', 'test.txt',
'testdir']
>>>
```

Figure 5: Create the Directory

3. Delete the directory



```
>>> uos.rmdir('testdir')
>>> uos.listdir()
['apn_cfg.json', 'test.py', 'wyc.mpy', 'wyc.py', 'usertest.mpy', 'test.txt']
>>>
```

Figure 6: Delete the Directory

NOTE

apn_cfg.json is the default script file.

4 Terms and Abbreviations

Table 3: Terms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
SDK	Software Development Kit