# QuecPython Timer Application Note

**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 | Randy/ Pawn | Creation of the document |
| 1.0.0 | 2020-11-09 | Randy/ Pawn | Preliminary |

# Contents

# 1 Introduction

This document takes Quectel EC100Y-CN module as an example to introduce how to use QuecPython class library API to develop and use a timer.

# 2 Timer Function

## 2.1. Basic Function of Timer

The timer is applied to multiple tasks. Quectel implements a simple function, the timer call function. Currently, the timer provided by Quectel realizes two mode, single and periodic, to call functions.

When the timer expires, an event will be triggered. With the callback function, the timer event can call a python function.

# 3 Timer Usage Example

Connect the EVB to the computer, and then refer to *Quectel_QuecPython_Basic_Operation_Guide* for next operations. The following takes the EC100Y-CN module as an example to explain how to use a timer.

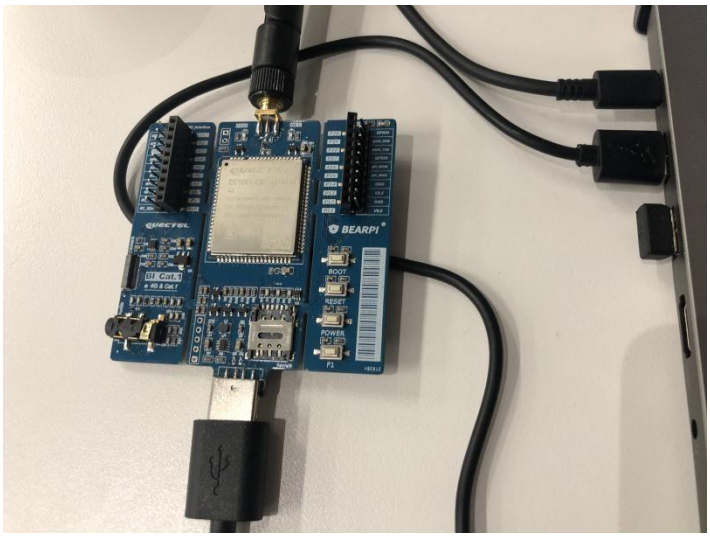

**Figure 1: The module connects to the Computer**

Create a *test.py* file, import timer class (included in the Machine module) of QuecPython into this file. The code to write a timer is as follows:

```
from machine import Timer

def func(args):
    print('###timer callback function###')

timer = Timer(Timer.Timer1)
timer.start(period=1000, mode=timer.PERIODIC, callback=func)
```

Upload *test.py* file to the EVB, see *Quectel_QuecPython_Basic_Operation_Guide* for details.

The result of the program is as follows:

```
>>> import example
>>> example.exec('test.py')
>>> ###timer callback function###
###timer callback function###
###timer callback function###
```

```
###timer callback function###
###timer callback function###
###timer callback function###
###timer callback function### ###timer callback function### timer.stop() 0
>>>
```

# 4 Timer in QuecPython

## 4.1. Constants in the Timer class

| Constants | Description |
|---|---|
| Timer.Timer0 | Timer 0 |
| Timer.Timer1 | Timer 1 |
| Timer.Timer2 | Timer 2 |
| Timer.Timer3 | Timer 3 |
| Timer.ONE_SHOT | Single mode, single timer execution |
| Timer.PERIODIC | Periodic mode, periodic timer execution |

## 4.2. Timer APIs

### 4.2.1. timer = Timer

This function creates a timer object. Before using the timer-related functions *timer.start* and *timer.stop*, you need to use this function to instantiate an object, that is, create a Timer object.

★ **Prototype**

```
timer = Timer(Timern)
```

★ **Parameter**

*Timern*:
A constant. Timer number. The timers supported by EC100Y-CN and EC600S-CN modules are: Timer0–Timer3.

★ **Return Value**

A timer object.

### 4.2.2. timer.start

This function starts a timer.

★ **Prototype**

```
timer.start(period, mode, callback)
```

★ **Parameter**

*period*:
Integer. Break-in period, unit: milliseconds.

*mode*:
A constant, the timer running mode, as follows:
Timer.ONE_SHOT Single mode, single timer execution
Timer.PERIODIC Periodic mode, periodic timer execution

*callback*:
Callback function, a function to be executed periodically.

★ **Return Value**

0 A timer is started successfully.
-1 It fails to start a timer.

### 4.2.3. timer.stop

This function stops a timer.

★ **Prototype**

```
timer.stop()
```

★ **Parameter**

None.

★ **Return Value**

0 A timer is stopped successfully.
-1 It fails to stop a timer.

# 5 Appendix

**Table 1: Terms and Abbreviations**

| Abbreviation | Description |
|---|---|
| API | Application Programming Interface |