

QuecPython 技术与资源综述

LTE Standard 模块系列

版本：1.0.0

日期：2020-11-11

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

目录

文档历史	2
目录	3
表格索引	5
图片索引	6
1 引言	7
2 软硬件技术综述	8
2.1. 硬件框架	8
2.2. 软件系统架构	9
2.3. QuecPython 系统特性	10
2.3.1. 简介	10
2.3.2. QuecPython SDK	11
3 资源介绍	14
3.1. 硬件资源介绍	14
3.1.1. 硬件连接图	14
3.1.2. 处理器	14
3.1.3. 空口资源	15
3.1.3.1. WAN 空口技术支持	15
3.1.4. 存储资源	15
3.1.4.1. 命令介绍	17
3.1.4.1.1. uos.statvfs	17
3.1.5. 外围资源	17
3.1.6. 支持的示例	18
3.2. 软件资源介绍	24
3.2.1. 启动流程	24
3.2.2. Python 脚本运行流程	25
3.2.3. QuecPython 内置模块	25
3.2.4. QuecPython 扩展类库	26
4 开发入门指南	28
4.1. 开发指导文档	28
4.2. 开发环境搭建	28
4.2.1. 系统版本	28
4.2.2. 小熊派开发板说明	28
4.2.3. 驱动安装	29
4.2.4. 固件下载	30
4.3. 运行第一个 QuecPython 程序	31
4.3.1. 下载 hello_world.py 程序到开发板	31
4.3.2. 执行 hello_world.py 程序	32
4.3.2.1. 手动执行	32
4.3.2.2. 自动运行	33

5 附录 A 参考文档及术语缩写 34

表格索引

表 1: WAN 空口技术支持	15
表 2: Flash 分区说明	16
表 3: 外围资源对应引脚号	17
表 4: Machine 硬件相关接口	18
表 5: 启动时间	24
表 6: QuecPython 内置模块	25
表 7: 扩展类库	26
表 9: 参考文档	34
表 9: 术语缩写	34

图片索引

图 1: QuecPython 硬件架构	8
图 2: QuecPython 软件技术架构	9
图 3: USB CDC 串口及其属性	11
图 4: QuecPython SDK 目录结构	13
图 5: QuecPython 硬件连接图	14
图 6: 启动流程	24
图 7: Python 脚本运行流程	25
图 8: 小熊派开发板	28
图 9: 串口名称与作用	29
图 10: 确认驱动安装成功	30
图 11: 确认固件版本	30
图 12: 固件下载界面	31
图 14: 脚本下载界面	32

1 引言

Python 是一门脍炙人口的编程语言，入门易、编写快、运行稳，不论是爬虫还是集群，都能看到它的身影，但是 Python 略显臃肿的体型，几乎和嵌入式开发无缘了。如果物联网也能使用 Python 开发呢？

现在，移远通信就可以使用 Python 进行开发，移远通信基于模块移植了 MicroPython 虚拟机，以实现 Python 在物联网中的开发与应用。

本文档以 EC100Y-CN 为例，主要介绍 EC100Y-CN 和 EC600S-CN QuecPython 软硬件系统资源及技术原理。

适用模块：

- EC100Y-CN（本文以该模块为例进行详细介绍）
- EC600S-CN

2 软硬件技术综述

2.1. 硬件框架

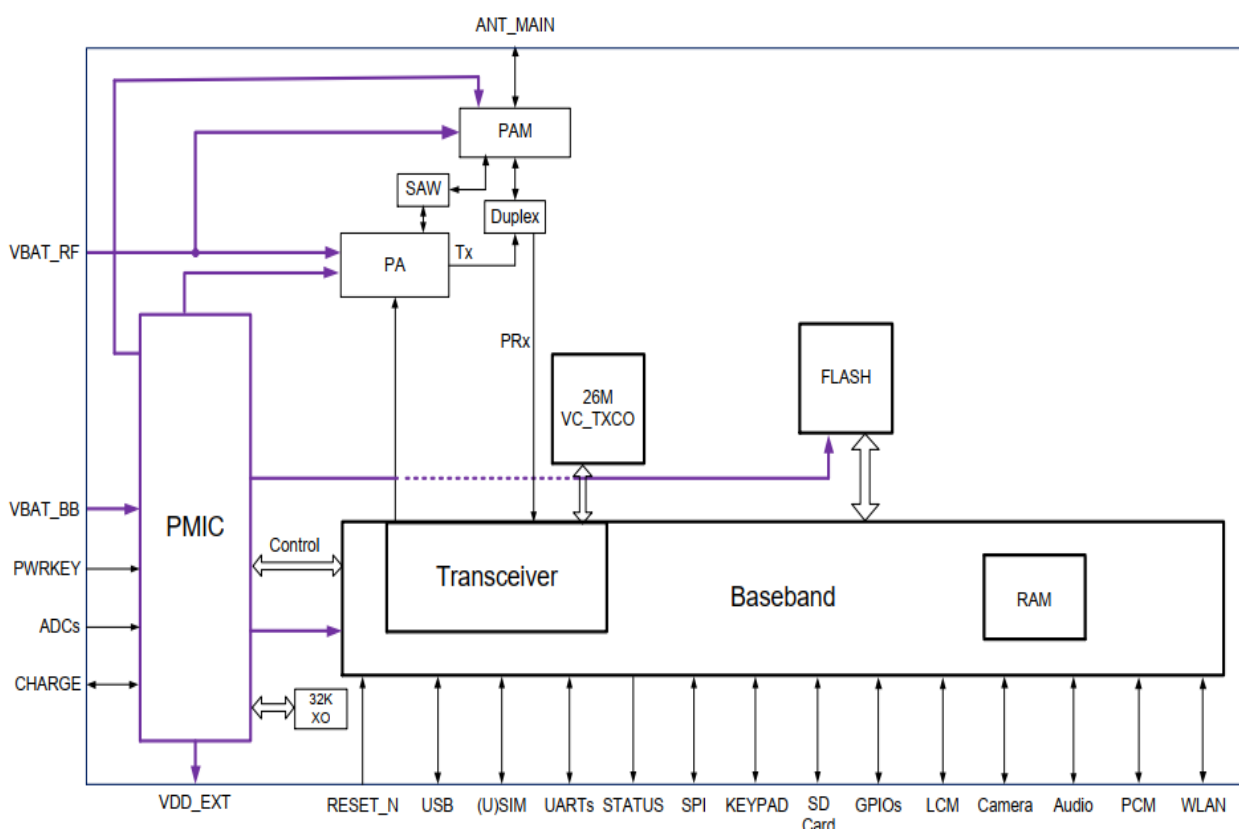


图 1: QuecPython 硬件架构

2.2. 软件系统架构

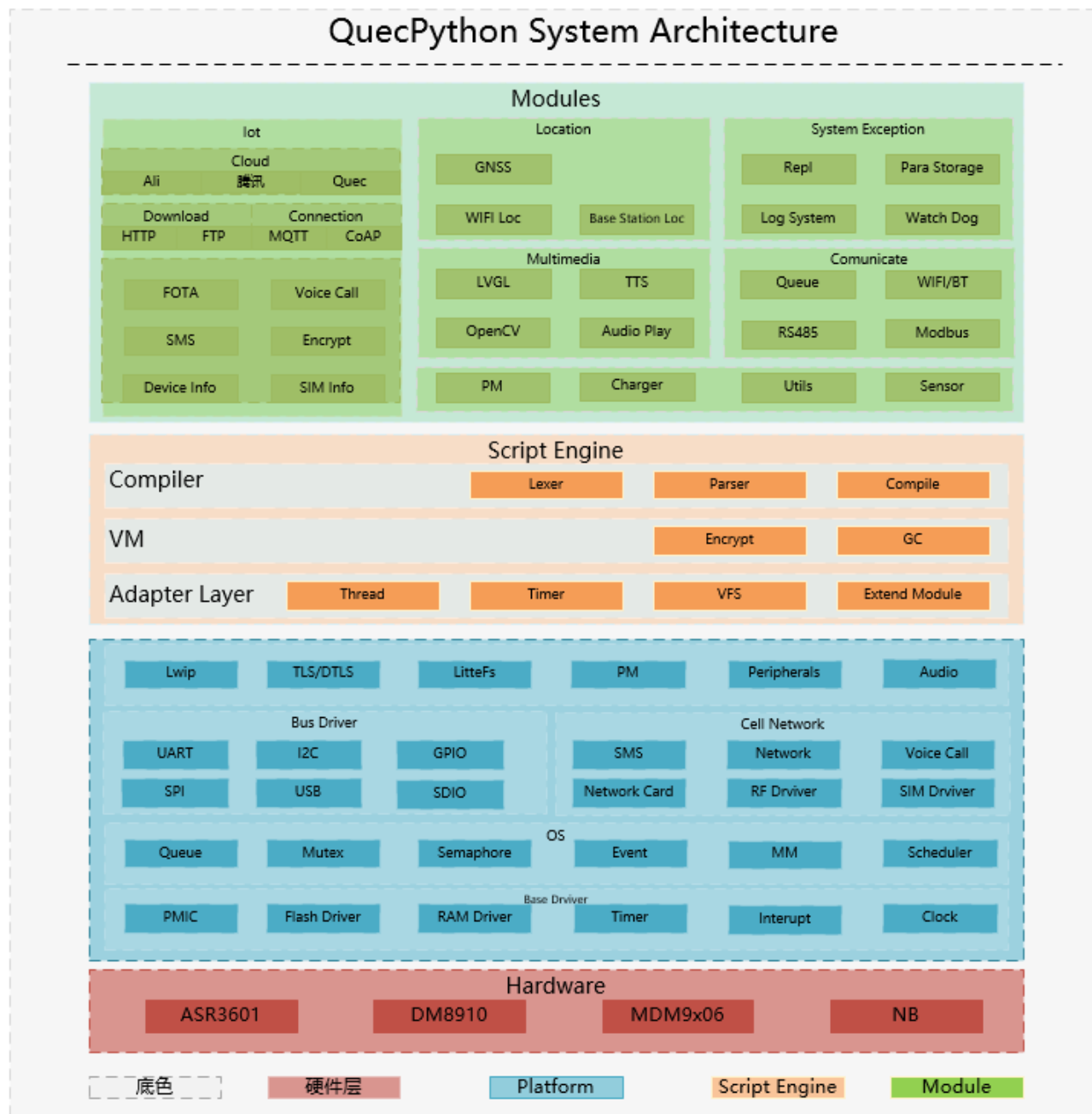


图 2: QuecPython 软件技术架构

备注

QuecPython 暂不支持 OpenCV、LVGL、Wi-Fi 及蓝牙功能。

2.3. QuecPython 系统特性

2.3.1. 简介

移远通信在模块中移植了 MicroPython，除了支持 MicroPython 功能外，还引入了通信模块相关功能，比如 Data call、MQTT、HTTP、Cloud、FOTA 等功能。

备注

MicroPython 是 Python 3 语言的精简高效实现，包括 Python 标准库的一小部分，经过优化可在微控制器和受限环境中运行。MicroPython 包含了诸如交互式提示、任意精度整数、关闭、列表解析、生成器、异常处理等高级功能，足够精简，适合运行在只有 256 KB 的代码空间和 16 KB 的 RAM 芯片上。MicroPython 旨在尽可能与普通 Python 兼容，让用户轻松将代码从桌面传输到微控制器或嵌入式系统。

QuecPython 用户代码执行方式有两种，REPL 和脚本文件：

- **REPL:** 交互式解释器，由 REPL 输入的脚本代码，经过词法分析、语法分析、编译，生成字节码，交由 QuecPython 虚拟机执行；
- **脚本文件:** 固件生成阶段将代码文件预处理成字节码，交由 QuecPython 虚拟机执行。

以上两种方式，均可通过 USB CDC 串口进行开发调试：

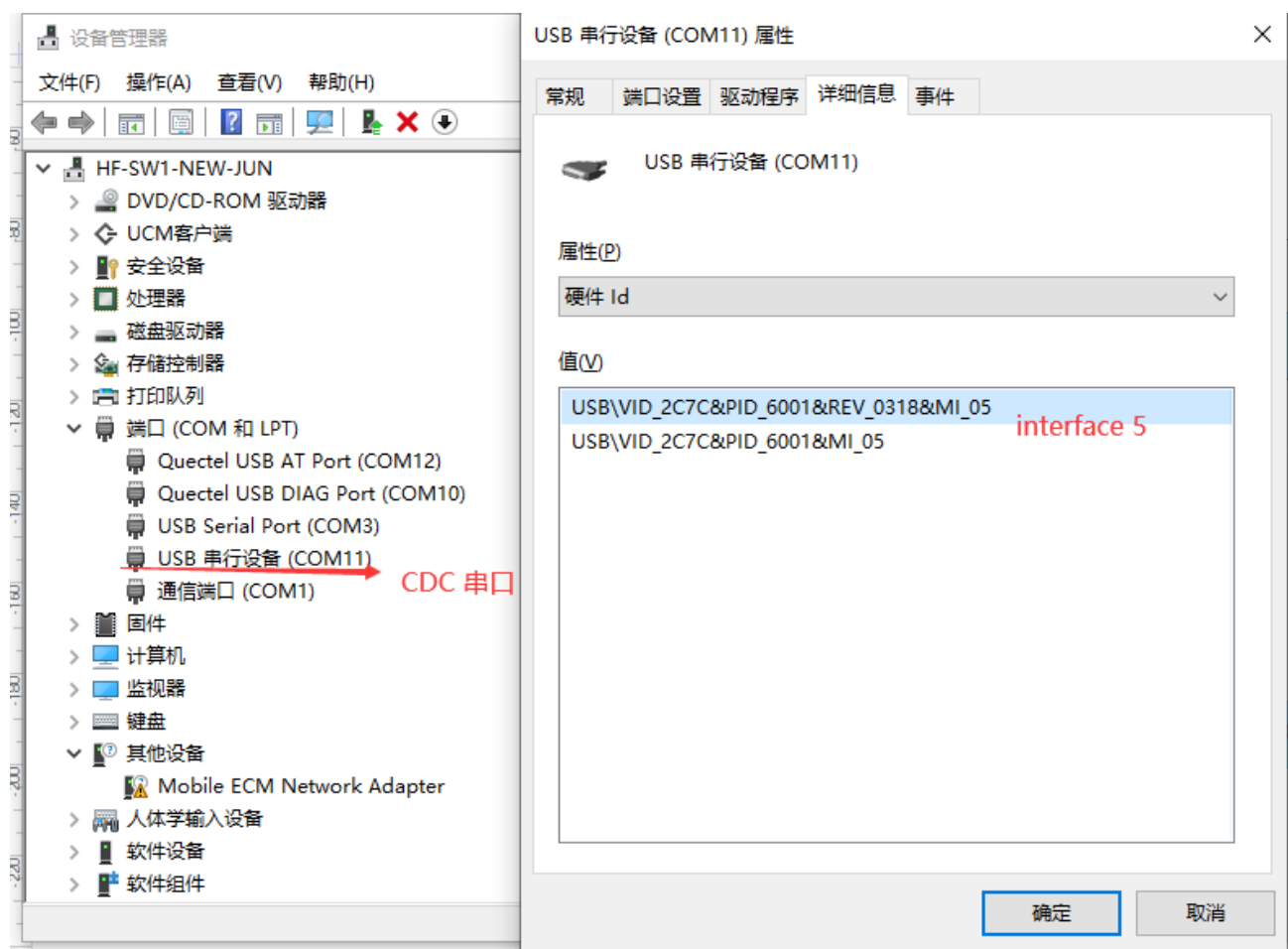


图 3: USB CDC 串口及其属性

串口调试界面如下所示:

```

QuecPython
Quecpython v1.12 on 2020-09-02; EC100Y with QUECTEL
Type "help()" for more information.
>>> uos.listdir()
['hello_world.py']
>>> import example
>>> example.exec('hello_world.py')
hello world!
>>> 

```

2.3.2. QuecPython SDK

QuecPython SDK 包含了与平台相关的文档、下载工具和各种功能示例源码，目录结构及说明如下：

```

README.md
boards
├── 小熊派-开发板
│   ├── Cat1_Core_RevA05.pdf
│   ├── Cat1_Core_RevB05_Gerber_生产文件.zip
│   └── Quectel_QuecPython_Cat1开发板使用说明_V1.3.pdf
cloud
├── ALiYun
│   └── aLiYun.py
├── QuecThing
└── TenCentYun
    └── TenCentYun.py
demo
├── README.md
├── aliyun
│   └── example_aliyun_file.py
├── common
│   ├── example_math_file.py
│   ├── example_random_file.py
│   ├── example_string_file.py
│   └── example_uos_file.py
├── dataCall
│   └── example_dataCall_callback_file.py
├── fota
│   └── example_fota_file.py
├── gpio
│   └── example_pin_file.py
├── http
│   ├── example_request_get_file.py
│   ├── example_request_post_file.py
│   └── example_request_ssl_file.py
├── I2C
│   └── example_i2c_file.py
├── json
│   └── example_json_file.py
├── log
│   ├── example_log_critical.py
│   ├── example_log_debug_file.py
│   ├── example_log_error_file.py
│   ├── example_log_info_file.py
│   └── example_log_warning_file.py
├── mqtt
│   └── example_mqtt_file.py
├── ntp
│   └── example_ntptime_file.py
├── pwm
│   └── example_pwm_file.py
├── socket
│   └── example_socket_file.py
├── TenCentyun
│   └── example_tencentyun_file.py
├── thread
│   └── example_thread_file.py
├── timer
│   └── example_timer_file.py
├── tts
│   └── example_tts_file.py
├── uart
│   └── example_uart_file.py
└── utime
    ├── example_utime_localtime_file.py
    ├── example_utime_mktime_file.py
    └── example_utime_sleep_file.py
    
```

```

- document
  Quectel QuecPython 类库API说明.md
  Quectel QuecPython_QPYcom工具使用说明_V1.0.pdf
  Quectel_QuecPython_Cat1开发板使用说明_V1.3.pdf
- firmware
  EC100YCNAAR01A01M16_OCPU_PY.zip
  README.md
- tools
  - USB驱动 USB Driver
    Quectel_ASR_Series_UMTS&LTE_Windows_USB_Driver_Customer_V1.0.3.zip
  - 下载工具 Download Tool
    QFlash_V4.20_CN.zip
    QLittlefs_tools.zip
  - 串口工具 Serial Tools
    QCOM_V1.6.zip
  - 图形化工具 QPYcom GUI tool
    QPYcom_V1.0.1.zip

```

图 4: QuecPython SDK 目录结构

关于 QuecPython SDK 介绍和使用方法，详见《Quectel QuecPython 类库 API 说明》。

备注

QuecPython 无需安装任何交叉编译工具链。Windows 10 仅需安装 Quectel USB 驱动（驱动位于 *tools* 目录下）即可开始开发。Windows 7 需要额外安装 USB CDC 驱动以解决安装交叉编译链的困扰。

3 资源介绍

3.1. 硬件资源介绍

3.1.1. 硬件连接图

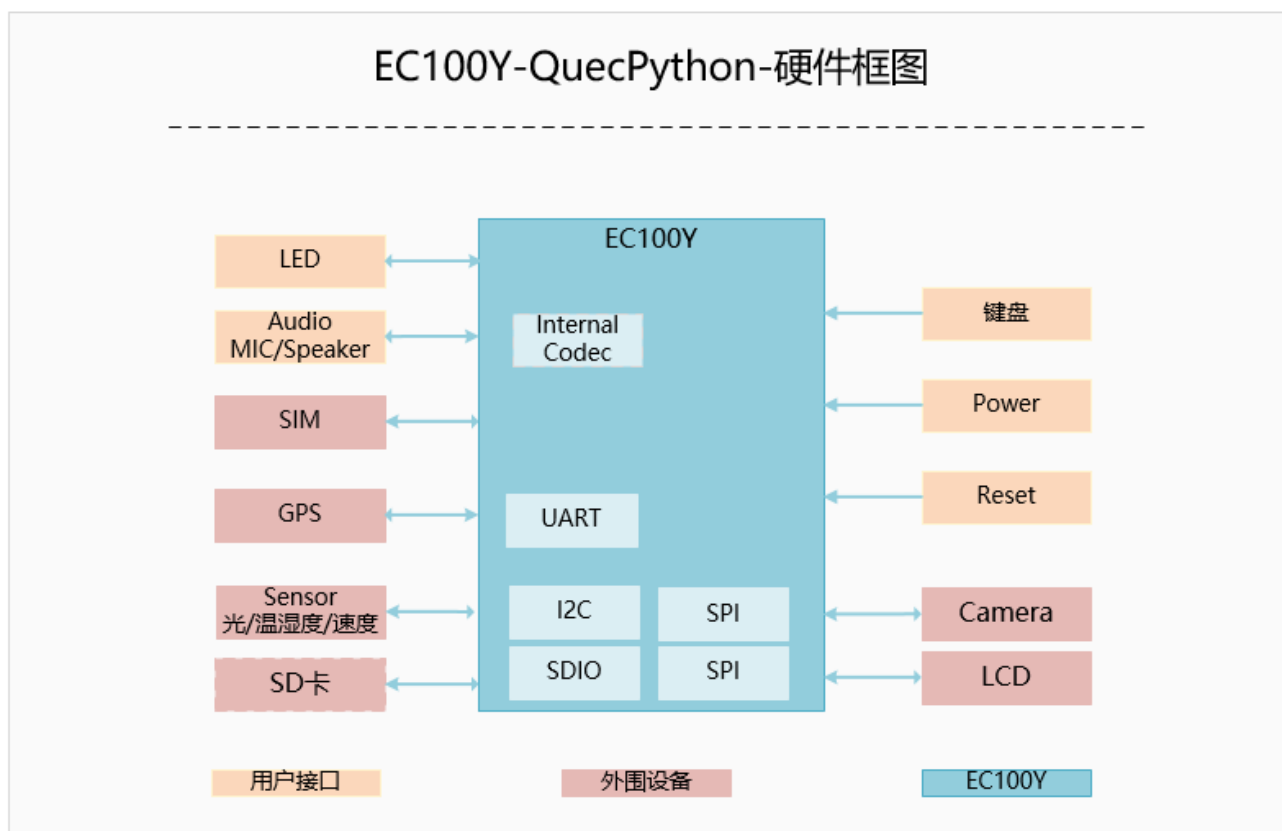


图 5: QuecPython 硬件连接图

3.1.2. 处理器

应用处理器: CPU 为 ARM Cortex-R5, 主频 624 MHz, 具有 32 KB 一级指令缓存和 32 KB 一级数据缓存

3.1.3. 空口资源

3.1.3.1. WAN 空口技术支持

表 1: WAN 空口技术支持

GSM	LTE
<ul style="list-style-type: none"> ● GSM/GRPS/EDGE Class 12 ● CAT-M1 R13 DL 1 Mbps / UL 1 Mbps 	<ul style="list-style-type: none"> ● DL 10 Mbps / UL 5 Mbps ● 支持 LTE ● 支持 VoLTE

3.1.4. 存储资源

EC100Y-CN 内置 16 MB Nor Flash 和 16 MB RAM，其中剩余可用空间如下：

- RAM: 3 MB 左右
- Flash: 4 MB 左右

可通过如下命令获取剩余 Flash 空间：

```
>>> uos.statvfs("main.py")
(4096, 4096, 1280, 1274, 1274, 0, 0, 0, 0, 255)
>>>
>>>
```

可通过如下命令获取 Python 剩余内存：

```
>>> gc.mem_free()
247904
>>>
```

系统剩余堆内存可通过 AT 命令获取。

备注

系统剩余 Flash 空间、Heap 及 GC 空间以实际查询值为准。

Flash 分区表的详细信息参见下表。因分区表可能会有调整，请以版本包中的分区信息为准。

表 2: Flash 分区说明

分区	大小 (K/M)	说明
External	16 M	/
Bootloader	144 K	启动固件
system	≈8 M	System 全部镜像
ptable	4 K	分区表信息
fwcerts	12 K	网络认证
Rd	64 K	/
Apn	32 K	/
cp	≈5 M	内核固件区；镜像大小自适应
Dsp	≈1.5 M	/
rfbin	20 K	/
logo	200 K	/
reserved	/	保留；镜像大小自适应
customer_app	≈1.5 M	用户固件区
customer_fs	4 M	用户文件系统固件区
customer_backup_fs	200 K	文件系统备份区
backup_restore_info	4 K	/
fota_param	12 K	升级参数区
fota_pkg	2.3 M	升级存储备份区
updater	128 K	升级固件区；内核
nvm	512K	/
erase_rd	64 K	/
quec_cfg	64 K	/
factory	128 K	/

3.1.4.1. 命令介绍

关于 `uos.statvfs` 和 `gc.mem_free` 的详细用法，详见《Quectel QuecPython 类库 API 说明》。

3.1.4.1.1. uos.statvfs

`uos.statvfs` 命令用于获取文件系统的状态信息。返回值格式为：(`f_bsize`, `f_frsize`, `f_blocks`, `f_bfree`, `f_bavail`, `f_files`, `f_ffree`, `f_favail`, `f_flag`, `f_namemax`)。

● 参数

参数名称	描述
<code>f_bsize</code>	文件系统块的大小。单位：字节
<code>f_frsize</code>	分块大小。单位：字节
<code>f_blocks</code>	文件系统数据块总数
<code>f_bfree</code>	可用块数
<code>f_bavail</code>	非超级用户可获取的块数
<code>f_files</code>	文件节点总数
<code>f_ffree</code>	可用文件节点数
<code>f_favail</code>	超级用户的可用文件节点数
<code>f_flag</code>	挂载标记
<code>f_namemax</code>	最大文件长度。单位：字节

3.1.5. 外围资源

外围资源的使用细节，详见《Quectel_EC100Y-CN_QuecOpen_硬件设计手册》。

表 3：外围资源对应引脚号

类型	组数	引脚号
UIM	1 组	57、58、59、60
USB	1 组，USB2.0 支持从模式	10、11、12

SDIO	1 组	29、30、31、32、33、34
UART	1 组	20、21
SPI	1 组	25、26、27、28
I2C	1 组	55、56
GPIO	5 组	22、23、178、199、204
PWM	1 组	16、17、18、19
ADC	2 组	39、81
LCD	1 组	21、22、23、24、25、26
Audio	1 组	2、3、4、41、42、82

3.1.6. 支持的示例

表 4: Machine 硬件相关接口

类	类功能	类方法	例程
硬件接口	PIN	GPIO 读写操作	<i>Pin.read()</i> <i>Pin.write(value)</i> demo/gpio/example_pin_file.py
	UART	UART 串口数据传输	<i>uart.any()</i> <i>uart.read(nbytes)</i> <i>uart.write(data)</i> <i>uart.close()</i> demo/uart/example_uart_file.py
	Timer	定时器	<i>timer.start(period, mode, callback)</i> <i>timer.stop()</i> demo/timer/example_timer_file.py
	ExtInt	配置 I/O 引脚在发生外部事件时中断	<i>extint.enable()</i> <i>extint.disable()</i> <i>extint.line()</i> http://qpy.quectel.com/wiki/#/zh-cn/api/?id=extint
	RTC	提供获取设置 RTC 时间方法	<i>rtc.datetime([year, month, day, week, hour, minute, second, microsecond])</i> http://qpy.quectel.com/wiki/#/zh-cn/api/?id=rtc
	WDT	看门狗（软狗）	<i>wdt.feed()</i> <i>wdt.stop()</i> demo/wdt/example_wdt_file.py
	IIC	设备之间通信的双线协议	<i>I2C.read(slaveaddress, addr, r_data, datalen, delay)</i> <i>I2C.write(slaveaddress, addr, data, datalen)</i> demo/I2C/example_i2c_file.py
UIO	包含其他类型的	<i>fd = uio.open(name, mode='r',</i>	demo/common/example

	stream（类文件） 对象和辅助函数	**kwarg) <i>fd.close()</i>	<i>_uos_file.py</i>
Data Call	数据拨号	<i>dataCall.start(profileIdx, ipType, apn, username, password, authType)</i> <i>dataCall.setApn(profileIdx, ipType, apn, username, password, authType)</i> <i>dataCall.setCallback(callback)</i> <i>dataCall.getInfo(profileIdx, ipType)</i>	<i>demo/dataCall/example_dataCall_callback_file.py</i>
基站定位	获取坐标信息	<i>cellLocator.getLocation(serverAddr, port, token, timeout, profileID)</i>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=celllocator- 基站定位
Audio	音频播放，支持 TTS、mp3 以及 AMR 文件播放	<i>tts.close()</i> <i>tts.play(priority, breakin, mode, str)</i> <i>tts.getVolume()</i> <i>tts.setVolume(vol)</i> <i>tts.getSpeed()</i> <i>tts.setSpeed(speed)</i> <i>tts.getState()</i> <i>tts.stop()</i> <i>aud.play(priority, breakin, filename)</i> <i>aud.stop()</i> <i>aud.getState()</i> <i>aud.getVolume()</i> <i>aud.setVolume(vol)</i>	<i>demo/tts/example_tts_file.py</i>
SIM 卡	SIM 卡操作，如查询 SIM 卡状态、ICCID、IMSI 等	<i>sim.getImsi()</i> <i>sim.getIccid()</i> <i>sim.getPhoneNumber()</i> <i>sim.getStatus()</i> <i>sim.enablePin(pin)</i> <i>sim.disablePin(pin)</i> <i>sim.verifyPin(pin)</i> <i>sim.unblockPin(puk, newPin)</i> <i>sim.changePin(oldPin, newPin)</i> <i>sim.readPhonebook(storage, start, end, username)</i> <i>sim.writePhonebook(storage, index, username, number)</i>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=sim-sim 卡
NET	配置和查询网络模式信息	<i>net.csqQueryPoll()</i> <i>net.getCellInfo()</i> <i>net.getConfig()</i> <i>net.getNetMode()</i> <i>net.getSignal()</i> <i>net.nitzTime()</i>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=net- 网络相关功能

		<code>net.operatorName()</code> <code>net.getState()</code> <code>net.getCi()</code> <code>net.getMnc()</code> <code>net.getMcc()</code> <code>net.getLac()</code> <code>net.getModemFun()</code> <code>net.setModemFun(function, rst)</code>	
FOTA	固件升级	<code>fota.write(bytesData)</code> <code>fota.flush()</code> <code>fota.verify()</code>	demo/fota/example_fota_file.py
UOS	包含文件系统访问和挂载构建, 实现了 CPython 模块相应模块的子集	<code>uos.remove(path)</code> <code>uos.chdir(path)</code> <code>uos.getcwd()</code> <code>uos.listdir([dir])</code> <code>uos.mkdir(path)</code> <code>uos.rename(old_path, new_path)</code> <code>uos.rmdir(path)</code> <code>uos.ilistdir([dir])</code> <code>uos.stat(path)</code> <code>uos.statvfs(path)</code> <code>uos.uname()</code> <code>uos.urandom(n)</code>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=uos-基本系统服务
gc	实现内存垃圾回收机制, 实现了 CPython 模块相应模块的子集	<code>gc.enable()</code> <code>gc.disable()</code> <code>gc.collect()</code> <code>gc.mem_alloc()</code> <code>gc.mem_free()</code>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=gc-内存碎片回收
socket	提供对 BSD 套接口字接口的访问	<code>usocket.getaddrinfo(host, port)</code> <code>socket.bind(address)</code> <code>socket.listen(backlog)</code> <code>socket.accept()</code> <code>socket.connect(address)</code> <code>socket.read([size])</code> <code>socket.readinto(buf, [, nbytes])</code> <code>socket.readline()</code> <code>socket.write(buf)</code> <code>socket.send(bytes)</code> <code>socket.sendall(bytes)</code> <code>socket.sendto(bytes, address)</code> <code>socket.recv(bufsize)</code> <code>socket.close()</code> <code>socket.setsockopt(level, optname, value)</code>	demo/socket/example_socket_file.py

		<code>socket.setblocking(flag)</code> <code>socket.settimeout(value)</code> <code>socket.makefile(mode='rb')</code>	
thread	提供创建新线程的方法，并提供互斥锁	<code>_thread.get_ident()</code> <code>_thread.stack_size()</code> <code>_thread.start_new_thread(function, args)</code> <code>_thread.allocate_lock()</code> <code>lock.acquire()</code> <code>lock.release()</code> <code>lock.locked()</code>	demo/thread/example_thread_file.py
urandom	提供了生成随机数的工具	<code>urandom.choice(obj)</code> <code>urandom.getrandbits(k)</code> <code>urandom.randint(start, end)</code> <code>urandom.random()</code> <code>urandom.randrange(start, end, step)</code> <code>urandom.seed(sed)</code> <code>urandom.uniform(start, end)</code>	demo/common/example_random_file.py
math	提供数学运算函数	<code>math.pow(x, y)</code> <code>math.acos(x)</code> <code>math.asin(x)</code> <code>math.atan(x)</code> <code>math.atan2(x, y)</code> <code>math.ceil(x)</code> <code>math.copysign(x, y)</code> <code>math.cos(x)</code> <code>math.degrees(x)</code> <code>math.e</code> <code>math.exp(x)</code> <code>math.fabs(x)</code> <code>math.floor(x)</code> <code>math.fmod(x, y)</code> <code>math.modf(x)</code> <code>math.frexp(x)</code> <code>math.isfinite(x)</code> <code>math.isinf(x)</code> <code>math.isnan(x)</code> <code>math.ldexp(x, exp)</code> <code>math.log(x)</code> <code>math.pi</code> <code>math.radians(x)</code> <code>math.sin(x)</code> <code>math.sqrt(x)</code> <code>math.tan(x)</code>	demo/common/example_math_file.py

		<i>math.trunc(x)</i>	
utime	获取当前时间和日期、测量时间间隔和延迟	<i>utime.localtime([secs])</i> <i>utime.mktime(date)</i> <i>utime.sleep(seconds)</i> <i>utime.sleep_ms(ms)</i> <i>utime.sleep_us(us)</i> <i>utime.ticks_ms()</i> <i>utime.ticks_us()</i> <i>utime.ticks_cpu()</i> <i>utime.ticks_diff(old, new)</i> <i>utime.time()</i>	demo/utime/example_utime_localtime_file.py demo/utime/example_utime_mktime_file.py demo/utime/example_utime_sleep_file.py
aLiYun	阿里云物联网套件客户端功能，目前的产品节点类型仅支持“设备”，设备认证方式支持“一机一密”和“一型一密”	<i>aLiYun(productKey, productSecret, DeviceName, DeviceSecret)</i> <i>aLiYun.setMqtt(clientID, clean_session, keepAlive)</i> <i>aLiYun.setCallback(sub_cb)</i> <i>aLiYun.subscribe(topic,qos)</i> <i>aLiYun.publish(topic,msg)</i> <i>aLiYun.start()</i>	demo/aliyun/example_aliyun_file.py
TXyun	腾讯云物联网套件客户端功能，目前的产品节点类型仅支持“设备”，设备认证方式支持“一机一密”和“一型一密”	<i>TXyun(productID, devicename, devicePsk, ProductSecret)</i> <i>TXyun.setMqtt(clean_session, keepAlive)</i> <i>TXyun.setCallback(sub_cb)</i> <i>TXyun.subscribe(topic,qos)</i> <i>TXyun.publish(topic,msg)</i> <i>TXyun.start()</i>	demo/TenCentyun/example_tencentyun_file.py
HTTP 服务	HTTP 客户端的相关功能函数	<i>request.get(url, data, json, headers)</i> <i>request.post(url, data, json, headers)</i> <i>request.put(url, data, json, headers)</i> <i>request.head(url, data, json, headers)</i> <i>request.patch(url, data, json, headers)</i> <i>request.delete(url, data, json, headers)</i> <i>response =request.get(url)</i> <i>response.content ()</i> <i>response.text ()</i> <i>response.json()</i> <i>response.close()</i>	demo/http/example_request_get_file.py demo/http/example_request_post_file.py demo/http/example_request_ssl_file.py
Log	系统日志记录，分级别日志工具	<i>log.basicConfig(level)</i> <i>log.getLogger(name)</i> <i>log.debug(tag, msg)</i> <i>log.info(tag,msg)</i>	demo/log/example_log_critical_file.py demo/log/example_log_debug_file.py

		<code>log.warning(tag,msg)</code> <code>log.error(tag,msg)</code> <code>log.critical(tag,msg)</code>	<code>demo/log/example_log_error_file.py</code> <code>demo/log/example_log_info_file.py</code> <code>demo/log/example_log_warning_file.py</code>
Umqtt 服务	提供创建 MQTT 客户端发布订阅功能	<code>MQTTClient(client_id, server, port=0, user=None, password=None, keepalive=0, ssl=False, ssl_params={})</code> <code>MQTTClient.set_callback(callback)</code> <code>MQTTClient.set_last_will(topic,msg,retain=False,qos=0)</code> <code>MQTTClient.connect(clean_session=True)</code> <code>MQTTClient.disconnect()</code> <code>MQTTClient.ping()</code> <code>MQTTClient.publish(topic,msg)</code> <code>MQTTClient.subscribe(topic,qos)</code> <code>MQTTClient.check_msg()</code> <code>MQTTClient.wait_msg()</code>	<code>demo/mqtt/example_mqtt_file.py</code>
NtpTime	时间同步	<code>ntptime.host</code> <code>ntptime.sethost(host)</code> <code>ntptime.settime()</code>	<code>demo/ntp/example_ntptime_file.py</code>
sys	提供了与 QuecPython 运行环境有关的函数和变量	<code>sys.argv</code> <code>sys.byteorder</code> <code>sys.implementation</code> <code>sys.maxsize</code> <code>sys.modules</code> <code>sys.platform</code> <code>sys.stdin</code> <code>sys.stdout</code> <code>sys.version</code> <code>sys.version_info</code> <code>sys.exit(retval=0)</code> <code>sys.print_exception(exc, file=sys.stdout)</code>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=sys-系统相关功能
ujson-Json	实现在 Python 数据对象和 JSON 数据格式之间进行转换的功能	<code>ujson.dump(obj, stream)</code> <code>ujson.dumps(dict)</code> <code>ujson.load(stream)</code> <code>ujson.loads(str)</code>	<code>demo/json/example_json_file.py</code>
ustruct	打包和解压原始数据类型	<code>ustruct.calcsize(fmt)</code> <code>ustruct.pack(fmt, v1, v2, ...)</code> <code>ustruct.unpack(fmt, data)</code>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=ustruct-打包

			<code>ustruct.pack_info(fmt, buffer, offset, v1, v2, ...)</code> <code>unstruct.unpack_from(fmt, data, offset=0)</code>	和解压原始数据类型
	Power 模块	提供关机、软件重启	<code>Power.powerDown()</code> <code>Power.powerRestart()</code> <code>Power.powerOnReason()</code> <code>Power.powerDownReason()</code> <code>Power.getVbatt()</code>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=power
misc	PWM 模块	PWM	<code>pwm = PWM(PWM.PWMn, highTime, cycleTime)</code> <code>pwm.open()</code> <code>pwm.close()</code>	demo/pwm/example_pwm_file.py
	ADC 模块	ADC	<code>adc.open()</code> <code>adc.read(ADCn)</code> <code>adc.close()</code>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=adc
modem 设备		设备信息获取	<code>modem.getDevImei()</code> <code>modem.getDevModel()</code> <code>modem.getDevSN()</code> <code>modem.getDevFwVersion()</code> <code>modem.getDevProductId()</code>	http://qpy.quectel.com/wiki/#/zh-cn/api/?id=modem-设备相关

3.2. 软件资源介绍

3.2.1. 启动流程



图 6：启动流程

表 5：启动时间

启动节点	启动时间	主要功能说明
Preboot	< 0.1 秒	检测电源开关键 检测是否进入下载模式
Bootloader	< 4 秒	检测是否进入 OTA 升级（暂不支持）

RTOS	< 6 秒	启动系统服务 注网并拨号
Python VM	< 7 秒	从文件系统读取并运行 Python 脚本

3.2.2. Python 脚本运行流程

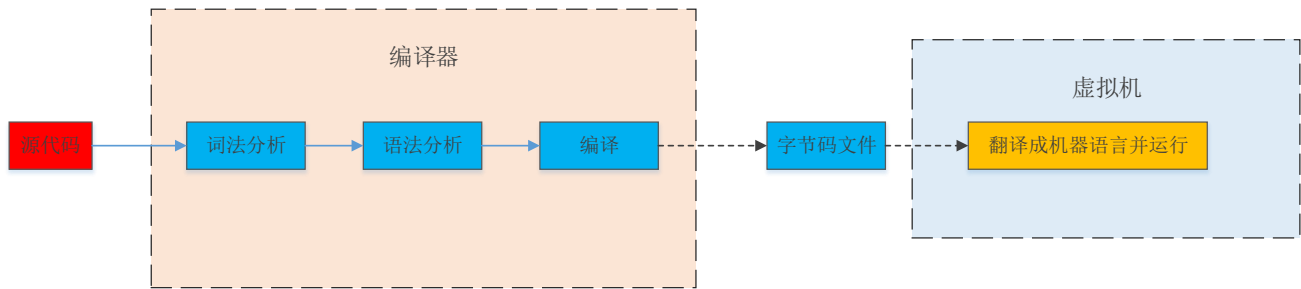


图 7: Python 脚本运行流程

系统运行后，QuecPython 作为 RTOS 的一个后台线程运行，从文件系统中读取并执行 Python 脚本。

3.2.3. QuecPython 内置模块

表 6: QuecPython 内置模块

内置模块名称	作用
uos	基本系统服务
gc	内存碎片回收
ubinascii	二进制与 ASCII 转换
ucollections	集合和容器类型
urandom	生成随机数
math	数学运算
usocket	Socket 模块
uio	输入输出流
ustruct	打包和解压原始数据类型
ujson	JSON 编码和解码

utime	与时间相关功能
sys	系统相关功能
uzlib	zlib 解压缩
_thread	多线程

3.2.4. QuecPython 扩展类库

表 7：扩展类库

模块名	作用
example	执行 Python 脚本
dataCall	数据拨号
cellLocator	基站定位
sim	SIM 卡相关
net	网络相关功能
fota	固件升级
audio	TTS 和音频文件播放
misc	重启、关机、PWM、ADC 相关功能
modem	设备相关
machine	硬件相关功能
aLiYun	阿里云服务
TenCentYun	腾讯云服务
request	HTTP
log	日志
umqtt	MQTT
ntptime	NTP 对时
pm	低功耗

ure

正则

4 开发入门指南

4.1. 开发指导文档

- 硬件开发指导：《Quectel_EC100Y-CN_QuecOpen_硬件设计手册》
- 软件开发指导：《Quectel_QuecPython_CAT1_开发板使用说明》

4.2. 开发环境搭建

4.2.1. 系统版本

系统版本要求为 Windows 7 以上版本。

4.2.2. 小熊派开发板说明

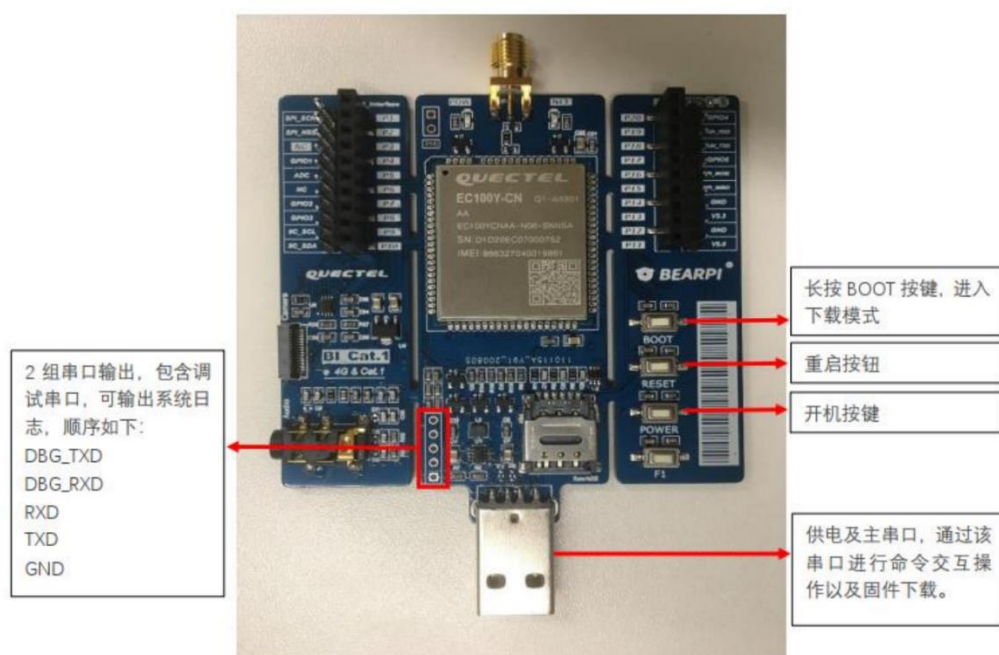



图 8：小熊派开发板

DBG_TXD、DBG_RXD 为调试信息输出端口，用户使用该端口时，需要将 DBG_TXD、DBG_RXD 以及 GND 连接到 USB To TTL 的转换器，通过转换器连接到电脑串口。

4.2.3. 驱动安装

在提供的软件工具中，找到如下驱动安装包并进行解压，双击运行 `setup.exe`，即可直接安装，安装完成点击“**Finish**”即可完成安装。

 Quectel_ASR_Series_UMTS<E_Windows_USB_Driver_Customer_V1.0.3.zip

驱动安装完成后，连接开发板，进入电脑的设备管理器，点击“端口(COM 和 LPT)”会出现如下三个端口（端口号可能与下图中不同，但是名称一致）：



图 9：串口名称与作用

然后，确认驱动是否安装成功。打开 QCOM 软件，如果可以正常发送 AT 命令并成功响应，则表示驱动安装成功。注意端口号要选择“**Quectel USB AT Port**”对应的端口号，波特率选择 115200，停止位 1 位，无奇偶校验，8 位数据位，无硬件控制流。详细信息如下图所示：

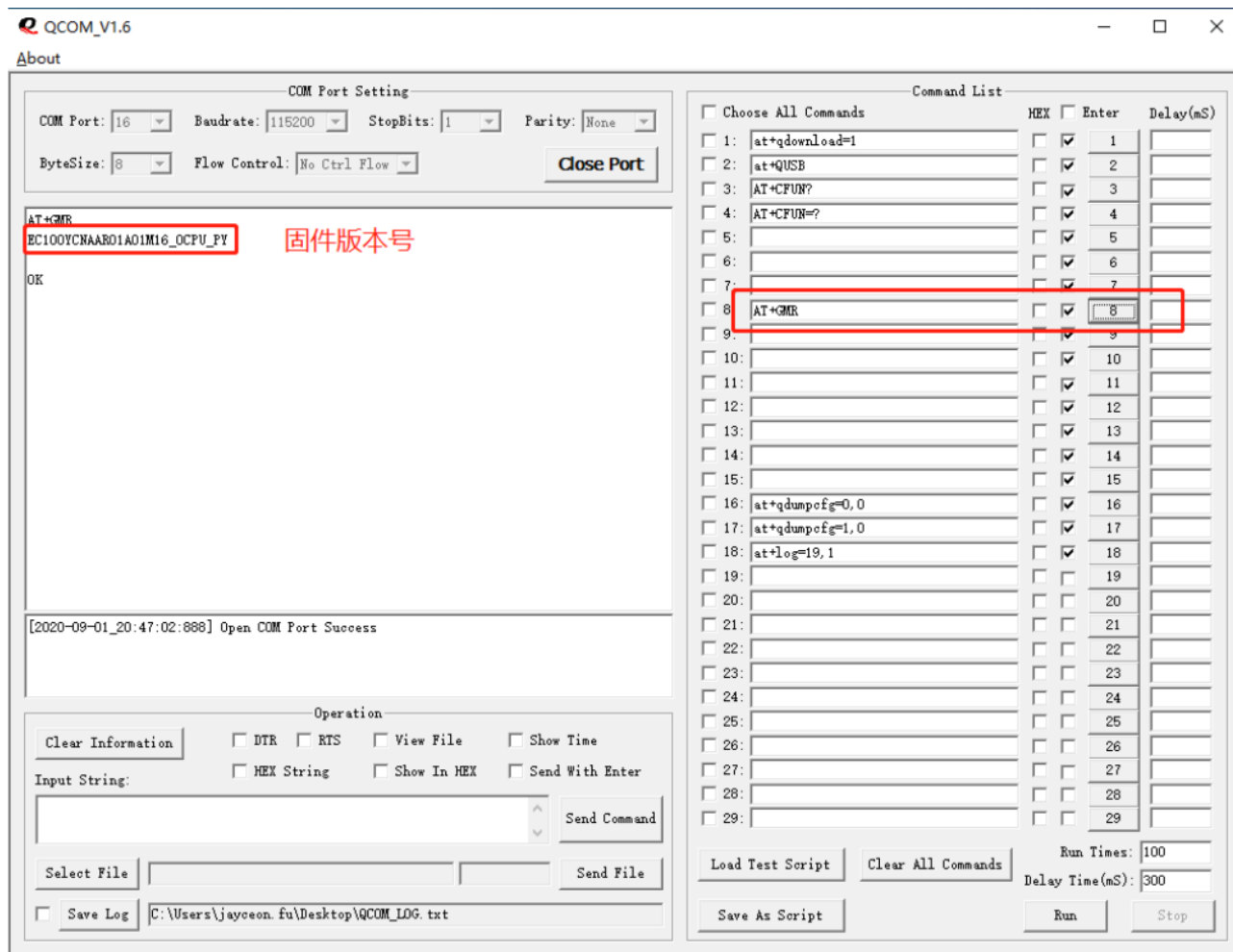


图 10：确认驱动安装成功

4.2.4. 固件下载

通过 QCOM 软件，发送 **AT+GMR**，通过返回值确认固件版本，如下所示：

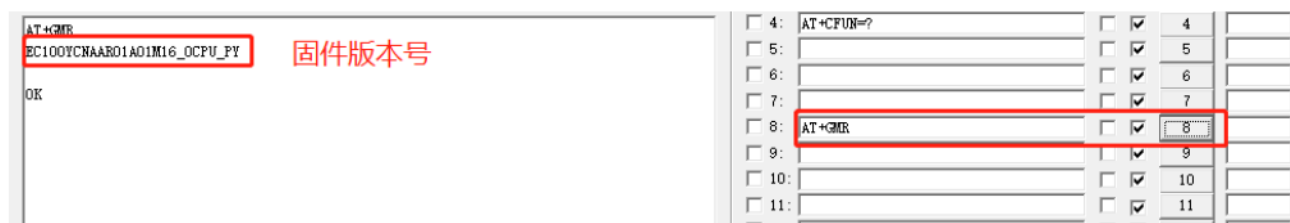


图 11：确认固件版本

若 **AT+GMR** 返回的固件号以“PY”结尾，表示该固件为 python 版，则直接跳过固件下载步骤。

若 **AT+GMR** 返回的固件号不以“PY”结尾，则需重新下载固件，下载方式如下：

- 步骤1: 在提供的软件工具中, 找到 QPYcom 下载工具压缩包, 解压后, 双击运行该软件;
- 步骤2: 根据需求, 创建用户项目 (点击“创建”按钮);
- 步骤3: 选择需要下载到模块的固件包;
- 步骤4: 点击“ ”按钮, 切换到“下载固件”模式;
- 步骤5: 点击“下载固件”模式;
- 步骤6: 进度条到 100%, 表示下载完毕。

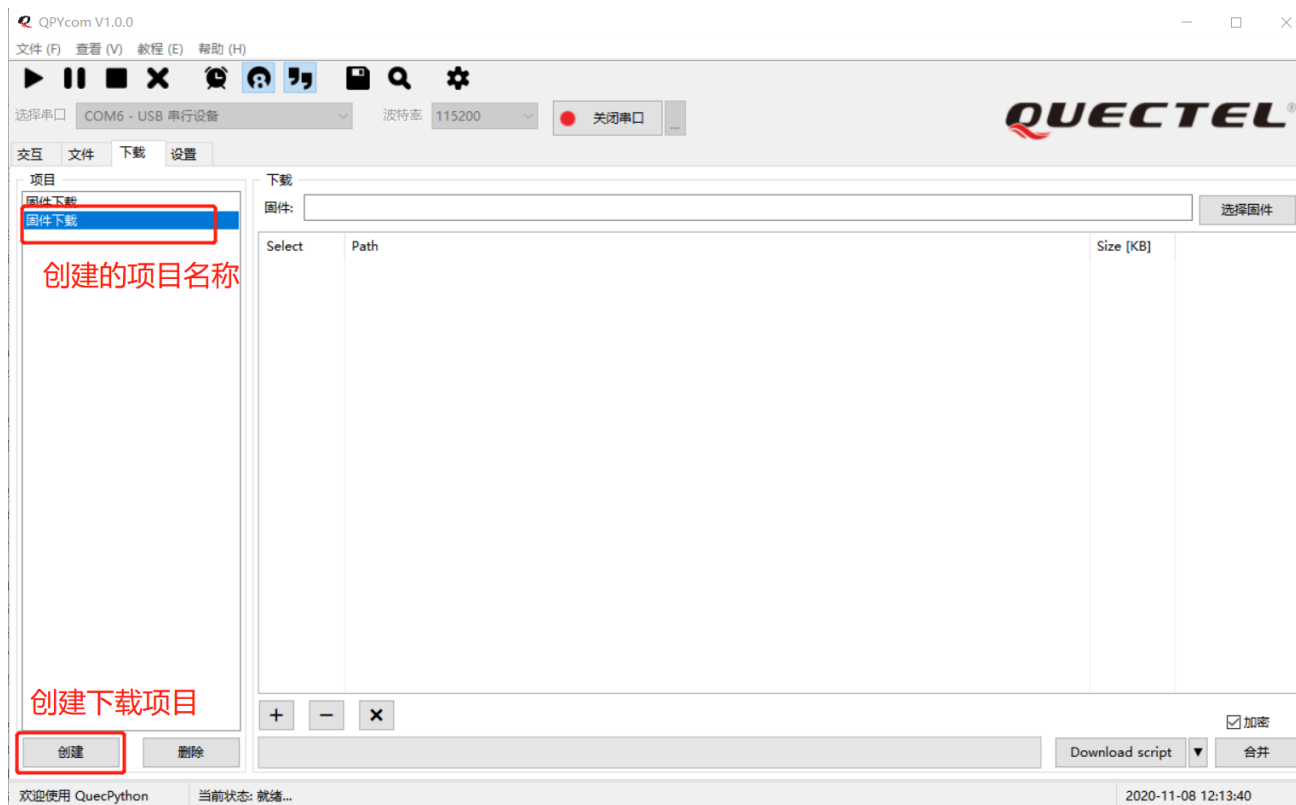


图 12: 固件下载界面

4.3. 运行第一个 QuecPython 程序

4.3.1. 下载 hello_world.py 程序到开发板

进入 SDK 工具包中的 `tools` 目录, 找到并解压 QPYcom 文件包, 进入解压后的目录找到 QPYcom.exe 工具, 通过该工具可以将 python 脚本文件发送到 EC100Y-CN 中。

- 步骤1: 根据需求, 创建用户项目 (点击“创建”按钮);
- 步骤2: 选择编写好的 hello_world.py 脚本;
- 步骤3: 点击“ ”按钮, 切换到“下载脚本”模式;

步骤4: 点击“下载脚本”按钮

步骤5: 进度条到 100%, 表示下载完毕。可点击左侧“文件”查看模块内文件详情。



图 13: 脚本下载界面

4.3.2. 执行 hello_world.py 程序

4.3.2.1. 手动执行

在 QPYcom 中, 连接 EC100Y-CN 主串口, 进入交互界面。首先通过 `uos.listdir()` 确认用户程序文件在当前目录下, 然后按如下步骤执行:

步骤1: 导入 example 模块, 该模块提供了 `exec()` 方法用来执行 python 脚本程序;

步骤2: 通过命令 `example.exec('hello_world.py')` 执行 hello_world.py 脚本。

结果如下所示:

```

Quecpython v1.12 on 2020-09-02; EC100Y with QUECTEL
Type "help()" for more information.
>>> uos.listdir()
['hello_world.py']
>>> import example
>>> example.exec('hello_world.py')
hello world!
>>> 

```

4.3.2.2. 自动运行

EC100Y-CN 支持上电自动执行用户代码。上电运行后，系统会查找名称为 *main.py* 的程序文件并自动执行该文件，因此若用户希望上电后自动运行代码，需将程序命名为 *main.py*，如果 *main.py* 中调用了其他源文件中提供的方法，那么需要将其他文件一起下载到模块中。下面仍以 *hello_world.py* 为例说明：

hello_world.py 文件提供 2 秒周期性打印 “hello world!” 字符串的方法；*main.py* 文件会调用 *hello_world.py* 中的方法。将两文件都下载到模块中，手动按一下开发板的 RESET 按键，系统启动后，重新连接主串口，电脑键盘按下 Enter 键，进入交互界面即可看到自动运行结果。

```

Connecting to COM25...
Connected.

hello world!
hello world!
hello world!
hello world!
hello world!

```

备注

每次系统重启后，都要断开主串口的连接，然后重新连接，否则看不到实验结果。

5 附录 A 参考文档及术语缩写

表 8: 参考文档

序号	文档名称	备注
[1]	Quectel QuecPython 类库 API 说明	QuecPython 类库 API 说明
[2]	Quectel_EC100Y-CN_QuecOpen_硬件设计手册	EC100Y-CN QuecOpen 硬件设计开发指导
[3]	Quectel_QuecPython_Cat1_开发板使用说明	QuecPython Cat 1 开发板使用说明

表 9: 术语缩写

术语	英文全称	中文全称
ADC	Analog-to-Digital Converter	模数转换器
API	Application Programming Interface	应用程序编程接口
CPU	Central Processing Unit	中央处理器
DL	Downlink	下行链路
FOTA	Firmware Upgrade Over-The-Air	固件空中升级
HTTP	Hypertext Transfer Protocol	超文本传输协议
LTE	(Long-Term Evolution) a 4G mobile communications standard.	长期演进
MQTT	Message Queuing Telemetry Transport	消息队列遥测传输
NTP	Network Time Protocol	网络时间协议
OTA	Upgrade Over-The-Air	空中升级
PWM	Pulse Width Modulation	脉冲宽度调制
RAM	Random Access Memory	随机存储器

RTOS	Real-Time Operating System	实时操作系统
SDK	Software Development Kit	软件开发工具包
SIM	Subscriber Identity Module	用户身份识别模块
TTS	Text To Speech	从文本到语音
UL	Uplink	上行链路
USB	Universal Serial Bus	通用串行总线
VoLTE	Voice (voice calls) over LTE. A standard high-speed wireless communication for mobile phones and data terminals, including Internet of things devices and wearables	长期演进语音承载
WAN	Wide Area Network	广域网 WAN
Wi-Fi	Wireless Fidelity	无线保真（技术）