

QuecPython

UART 使用说明

LTE 系列

版本：UART 开发使用说明_V1.0

日期：2020-12-29

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。由于客户操作不当而造成的人身伤害或财产损失，本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。

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
1.0	2020-12-29	唐大为	初始版本

目录

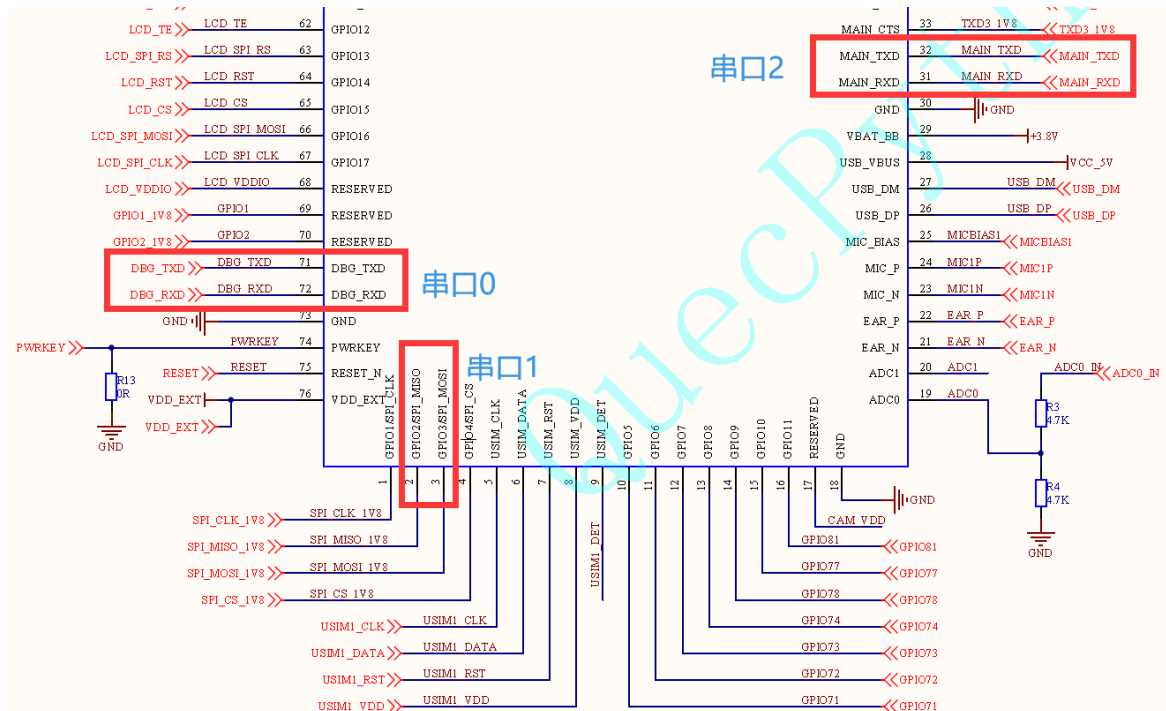
1	引言.....	- 5 -
2	硬件描述.....	- 5 -
3	软件设计.....	- 6 -
3.1	创建 UART 对象.....	- 6 -
3.2	查询串口未读字节数.....	- 6 -
3.3	读取数据.....	- 6 -
3.4	发送数据.....	- 7 -
3.5	关闭串口.....	- 7 -
4	交互操作.....	- 7 -
5	下载验证.....	- 8 -
5.1	软件代码.....	- 8 -
5.2	硬件连接.....	- 9 -
5.3	运行效果.....	- 9 -

1 引言

文档主要基于 EC600S 介绍如何使用 QuecPython_UART，UART 作为一种通用串行数据总线，用于异步通信，可实现全双工传输和接收。在嵌入式设计中，UART 用来与 PC 进行通信，包括监控调试器和其它器件。通过本文你将了解到 EC600S_UART 的所有设置参数及使用方法。

2 硬件描述

目前开放共 3 路 UART，各路 UART 对应模组的 PIN 脚如下图所示：



3 路 UART 对应开发板的接口如下图所示：



3 路串口对应关系列表如下：

串口名	模组引脚名	模组引脚号
调试串口	DBG_RXD	72
	DBG_TXD	71
辅助串口	GPIO2/SPI_MISO	2
	GPIO3/SPI_MOSI	3
主串口	MAIN_DTR	39
	MAIN_RXD	31
	MAIN_TXD	32
	MAIN_CTS	33
	MAIN_RTS	34
	MAIN_DCD	48
	MAIN_RI	40

3 软件设计

3.1 创建 UART 对象

uart = UART(UART.UARTn, baudrate, databits, parity, stopbits, flowctl)。创建 UART 对象接口参数介绍如下：

参数	类型	说明
UARTn	int	端口号 UART0 - DEBUG PORT UART1 - BT PORT UART2 - MAIN PORT
baudrate	int	波特率，常用波特率都支持，如 4800、9600、19200、38400、57600、115200、230400 等
databits	int	数据位（5~8）
parity	int	奇偶校验（0 - NONE，1 - EVEN，2 - ODD）
stopbits	int	停止位（1~2）
flowctl	int	硬件控制流（0 - FC_NONE，1 - FC_HW

3.2 查询串口未读字节数

uart.any(): 返回接收缓存器中有多少字节的数据未读，无参数。

返回值：指令执行成功或者不存在未读数据都返回 0，否则返回未读字节数。

3.3 读取数据

uart.read(nbytes): 从串口读取数据，建议先执行上个命令，查询未读数据的字节数，然后读取数据时加上需要读取的字节数。

参数介绍如下:

参数	类型	说明
nbytes	int	要读取的字节数

返回值: 返回读取的数据

3.4 发送数据

uart.write(data), 用于从模组向外发送数据, 参数介绍如下:

参数	类型	说明
data	string	发送的数据

返回值: 返回发送的字节数

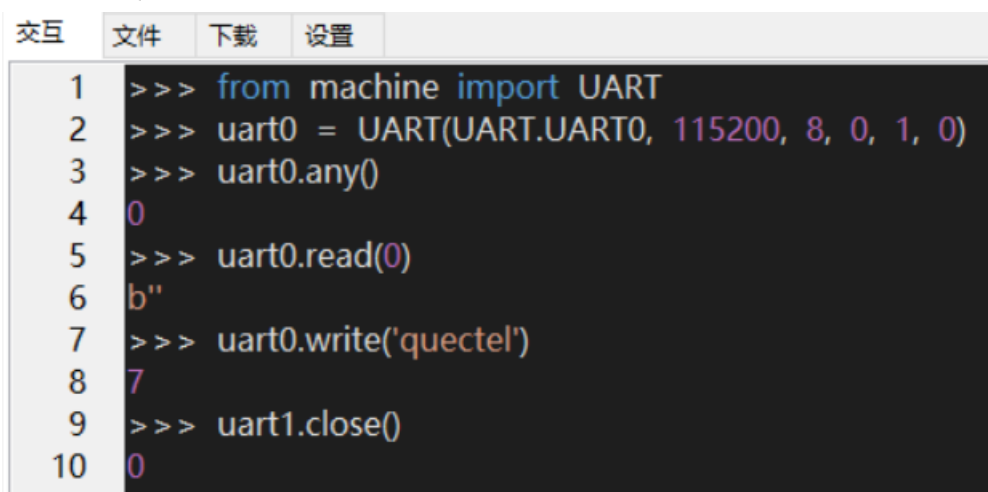
3.5 关闭串口

uart.close(): 关闭串口, 无参数。

返回值: 成功返回整型 0, 失败返回整型-1

4 交互操作

使用 QPYcom 工具和模组进行交互, 下面实例是基于 UART0, UART1 和 UART2 操作类似。



```

交互  文件  下载  设置
1  >>> from machine import UART
2  >>> uart0 = UART(UART.UART0, 115200, 8, 0, 1, 0)
3  >>> uart0.any()
4  0
5  >>> uart0.read(0)
6  b''
7  >>> uart0.write('quectel')
8  7
9  >>> uart1.close()
10 0
  
```

注意:

1. from machine import UART 即为让 UART 模块在当前空间可见。
2. 只有 from machine import UART 模块, 才能使用 UART 内的函数和变量。
3. 上述操作没有连接任何外设, 仅作为熟悉指令参考

5 下载验证

5.1 软件代码

配套 demo 的参考代码为文档同目录下的 example_uart_file.py 文件。下载.py 文件到模组运行，代码如下：

```
import _thread # 导入线程模块
import utime # 导入定时模块
import log # 导入 log 模块
from machine import UART # 导入 UART 模块
# 测试该示例代码需要配置 uart
# 设置日志输出级别
log.basicConfig(level=log.INFO)
uart_log = log.getLogger("UART")
state = 1
def uartWrite():
    count = 10
    # 配置 uart
    uart = UART(UART.UART1, 115200, 8, 0, 1, 0)
    while count:
        write_msg = "Hello count={}".format(count)
        # 发送数据
        uart.write(write_msg)
        uart_log.info("Write msg : {}".format(write_msg))
        utime.sleep(1)
        count -= 1
    uart_log.info("uartWrite end!")
def UartRead():
    global state
    uart = UART(UART.UART1, 115200, 8, 0, 1, 0)
    while 1:
        # 返回是否有可读取的数据长度
        msgLen = uart.any()
        # 当有数据时进行读取
        if msgLen:
            msg = uart.read(msgLen)
            # 初始数据是字节类型 (bytes), 将字节类型数据进行编码
            utf8_msg = msg.decode()
            # str
            uart_log.info("UartRead msg: {}".format(utf8_msg))
        else:
            continue
    state = 0
```



```
def run():
    # 创建一个线程来监听接收 uart 消息
    _thread.start_new_thread(UartRead, ())
if __name__ == "__main__":
    uartWrite()
    run()
    while 1:
        if state:
            pass
        else:
            break
```

5.2 硬件连接

使用 type-c 给模块供电，UART 与 TTL 转 USB 模块的连接如下表（仅供参考）：

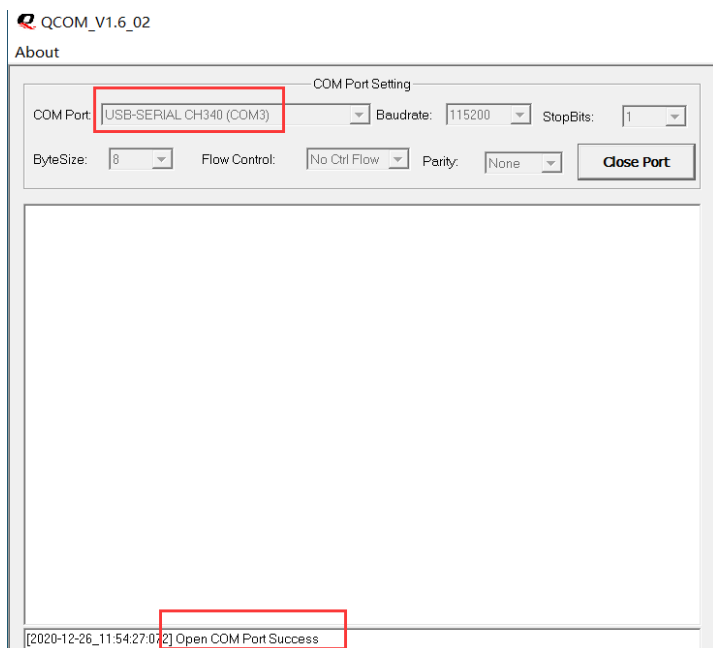
模块 UART_pin 脚	TTL 转 USB 模块
SPI_MISO	Tx
SPI_MOSI	Rx
GND	GND

硬件连接展示图如下图所示：



5.3 运行效果

- (1) 打开 QCOM，连接 TTL 转 USB 对应的端口，如下图所示：



- (2) 打开 QPYcom 运行 uart_file.py, 可在 QCOM 查看到 UART write 数据, 现象如下截图红色部分
- (3) 然 QCOM 上写 quectel 两次, 可在 QPYcom 上查看读取的数据。现象如下截图蓝色部分

