

QuecPython 总线用户指导

LTE Standard 模块系列

版本：1.0.0

日期：2020-11-10

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
-	2020-11-10	Rivern/Kenney	文档创建
1.0.0	2020-11-10	Rivern/Kenney	临时版本

目录

文档历史	2
图片索引	4
1 引言	5
2 总线基本功能和使用方法介绍	6
2.1. ADC 数模转换	6
2.2. UART	8
2.3. SPI 通信	9
2.4. I2C 通信	10
3 附录 A 术语缩写	13

图片索引

图 1: ADC 输入引脚.....	7
图 2: ADC 通道电压变化	8
图 3: UART 硬件连接.....	9
图 4: UART API 示例代码	9
图 5: SPI 硬件连接.....	10
图 6: I2C 硬件连接.....	11
图 7: I2C 连接光照传感器	12

1 引言

所谓总线（Bus），是指计算机设备和设备之间传输信息的公共数据通道。总线是连接计算机硬件系统内多种设备的通信线路，它的一个重要特征是由总线上的所有设备共享，可以将计算机系统内的多种设备连接到总线上。如果是某两个设备或设备之间专用的信号连线，就不能称之为总线。

本文介绍四种外围总线基本功能和基本使用方法，主要涉及 ADC、UART、SPI、I2C。

适用模块：

- EC100Y-CN（本文以该模块为例进行介绍）
- EC600S-CN

2 总线基本功能和使用方法介绍

2.1. ADC 数模转换

数字信号和模拟信号转换器 ADC，称为数模转换器，CPU 本身是数字的，但是外部的一些变量是模拟的，所以需要利用数字技术处理外部模拟的物理量。模拟信号，是一个连续的信号，现实生活中的时间，电压，高度等就是模拟信号，反应在数学里就是无限细分的值。

数模转换就是把模拟信号按照一定精度进行采样，变成有限多个数字量，这个过程就是数模转换，数字化之后就可以在计算机中用数字来描述模拟量，是计算机技术的基础，计算机所有参与运算的都是数字量，如果参与计算的有模拟量，就需要使用数模转换器将模拟量转换为数字量来参与运算，同样，也可以通过使用有积分和微分效果的器件来将数字信号转换为模拟信号。

QuecPython 开发板中 ADC 输入引脚如下图所示。

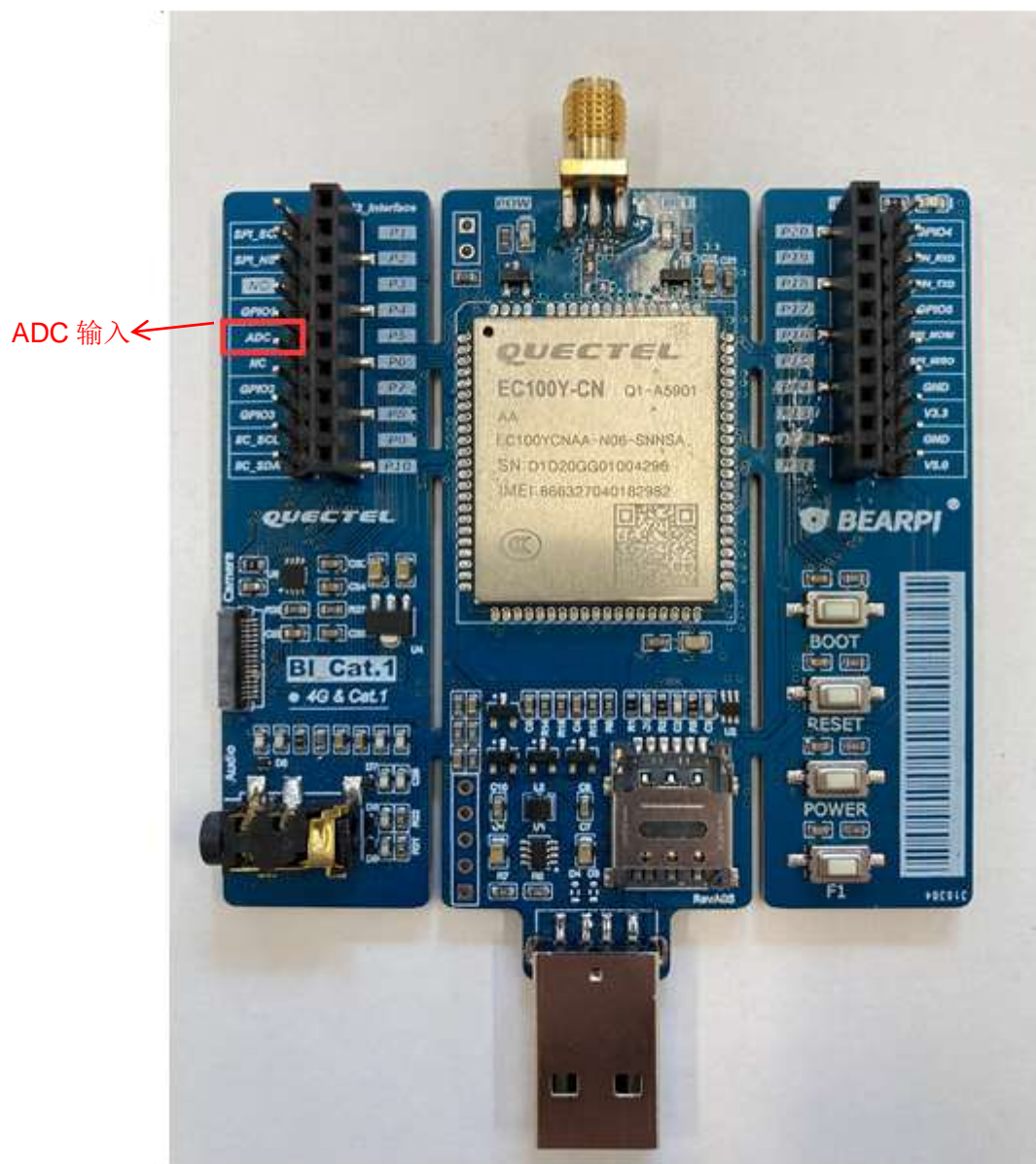


图 1: ADC 输入引脚

将 GPIO1 串口和 ADC 串口连接，在 QuecPython 中通过 *misc* 模块 *ADC* 类读取通道电压值，通过 *machine* 模块 *Pin* 类中 *Pin.write(value)* 方法设置 PIN 脚电平，详细使用方法和 API 接口说明见《Quectel_QuecPython_类库 API 说明》。

```
from machine import Pin
from misc import ADC
adc = ADC
adc.open()
```



```
gpio1 = Pin(Pin.GPIO1, Pin.OUT, Pin.PULL_DISABLE, 0)
gpio1.read() # 获取 gpio 的当前高低状态
gpio1.write(1) # 设置 gpio1 输出高
adc.read(ADC.ADC0)
gpio1.write(0) # 设置 gpio1 输出低
adc.read(ADC.ADC0)
```

命令行运行结果可见ADC通道电压变为1.8 V，如图所示：

```
>>> gpio.write(0)
0
>>> adc.read(ADC.ADC0)
0
>>> gpio.write(1)
0
>>> adc.read(ADC.ADC0)
1803
>>> []
```

图 2：ADC 通道电压变化

2.2. UART

UART 是一种串行异步收发协议，应用十分广泛。UART 工作原理是将数据的二进制位一位一位的进行传输。在 UART 通讯协议中信号线上的状态位高电平代表 1，低电平代表 0。当然两个设备使用 UART 串口通讯时，必须先约定好传输速率和一些数据位。

硬件连接比较简单，仅需要 3 条线，注意连接时若两个设备 UART 电平范围不一致请做电平转换后再连接。

- TX：发送数据端，要接对面设备的 RX；
- RX：接收数据端，要接对面设备的 TX；
- GND：保证两设备共地，有统一的参考平面。

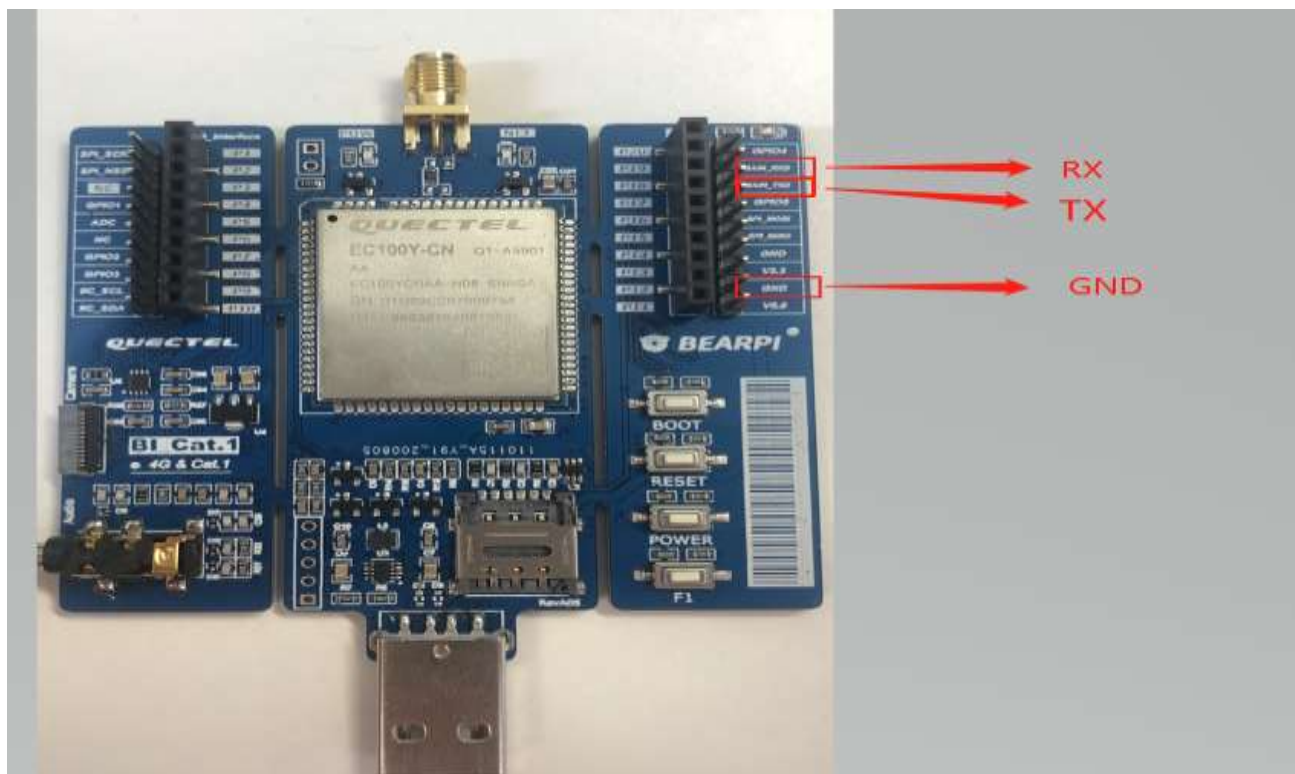


图 3: UART 硬件连接

在 EC100Y-CN 模块上使用 UART 串口传输数据时，需将发送数据端 TX 与对面的 RX 相连，将接受数据端 RX 与对面的 TX 相连，在 QuecPython 中通过 machine 模块 UART 类可以实现串口数据传输功能，详细使用方法和 API 接口说明见《Quectel_QuecPython 类库 API 说明》。

```
>>>
>>> from machine import UART
>>> uart1 = UART(UART.UART2,9600,8,0,1,0)
>>> uart1.any()
8192
>>> uart1.read(1024)
b'$GPGGA,000102.262,,,,,0,0,,,M,M,,*4D\r\n$GPGLL,,,,,000102.262,V,N*7F\r\n$GPGSA,A,1,,,,,,,,,,,,,*1E\r\n$
00103.262,V,N*7E\r\n$GPGSA,A,1,,,,,,,,,,,,,*1E\r\n$GPGSV,1,1,00*79\r\n$GPRMC,000103.262,V,,,,,0.00,0.00,060
GSV,1,1,00*79\r\n$GPRMC,000104.262,V,,,,,0.00,0.00,060180,,N*41\r\n$GPVTG,0.00,T,M,0.00,N,0.00,K,N*32\r\n$
80,,N*40\r\n$GPVTG,0.00,T,M,0.00,N,0.00,K,N*32\r\n$GPGGA,000106.262,,,,,0,0,,,M,M,,*49\r\n$GPGLL,,,,,0001
GGA,00'
```

图 4: UART API 示例代码

2.3. SPI 通信

SPI 协议是串行外围设备接口，是一种高速全双工的通信总线。SPI 总线包含 4 条总线，分别为 SS、SCK、MOSI、MISO。

(1) **SS**: 片选信号线, 当有多个 SPI 设备与 MCU 相连时, 每个设备的这个片选信号线是与 MCU 单独的引脚相连的, 而其他的 SCK、MOSI、MISO 线则为多个设备并联到相同的 SPI 总线上, 低电平有效。

(2) **SCK**: 时钟信号线, 由主通信设备产生。不同的设备支持的时钟频率不一样, 如 STM32 的 SPI 时钟频率最大为 $f_{PCLK} / 2$ 。

(3) **MOSI**: 主设备输出 / 从设备输入引脚。主机的数据从这条信号线输出, 从机由这条信号线读入数据, 即这条线上数据的方向为主机到从机。

(4) **MISO**: 主设备输入 / 从设备输出引脚。主机从这条信号线读入数据, 从机的数据则由这条信号线输出, 即在这条线上数据的方向为从机到主机。

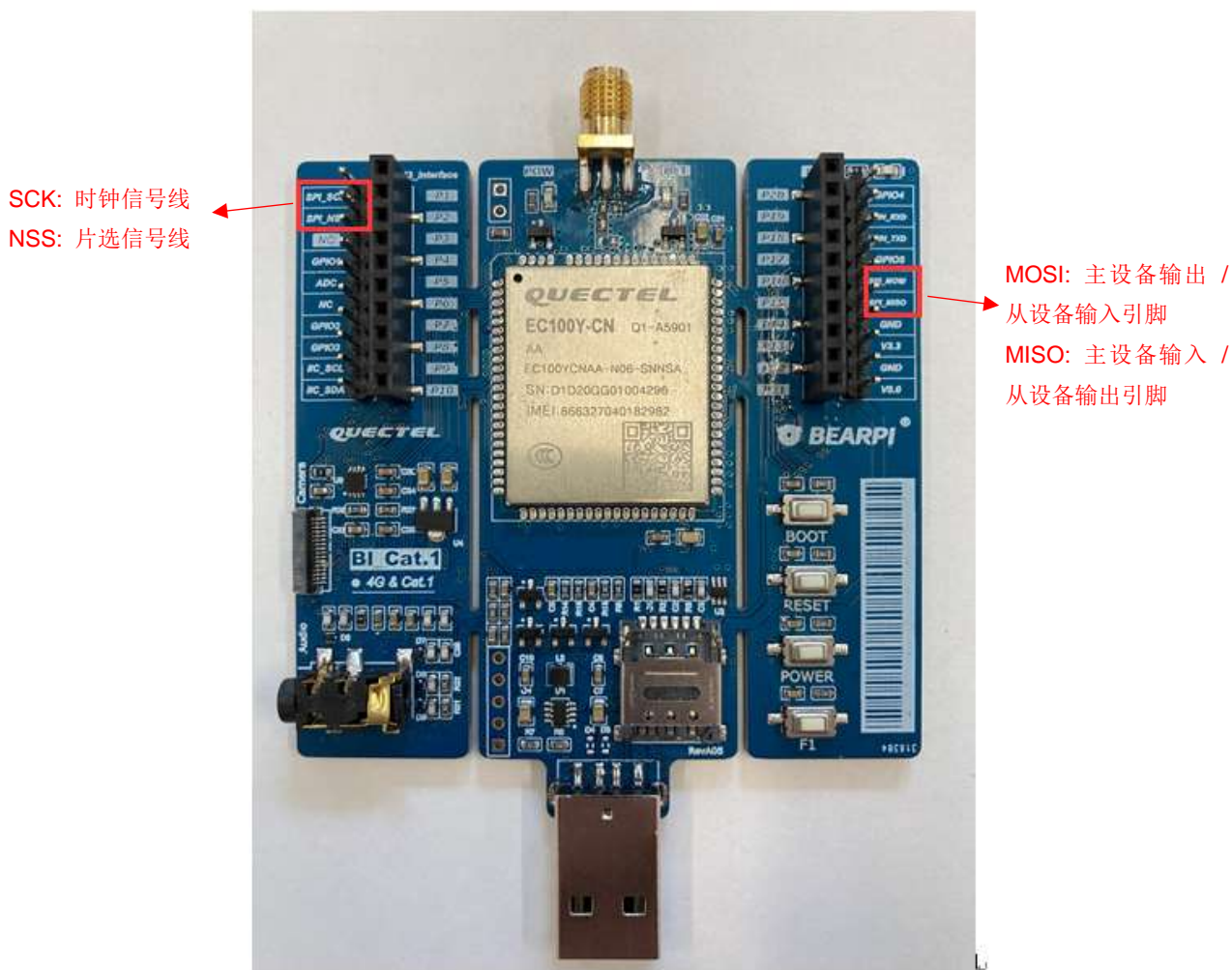


图 5: SPI 硬件连接

2.4. I2C 通信

I2C 接口只有两根线, SCL 和 SDA:

- **SCL**: 传输时钟信号, 由主设备向从设备传输时钟信号。

- SDA: 传输数据信号，主从设备之间相互传递数据的通道。

I2C 属于串行通信，数据以 bit 为单位在 SDA 线上串行依次传输，同步工作状态，主从设备工作在同一个时钟频率下，通过 SCL 线同步时钟，I2C 传输电平信号，不需要很高的速度，通信双方距离很近，所以不需要差分信号来抗干扰，I2C 通常用在同一块板子上的两个 IC 之间的通信，数据量不大且速度较低。

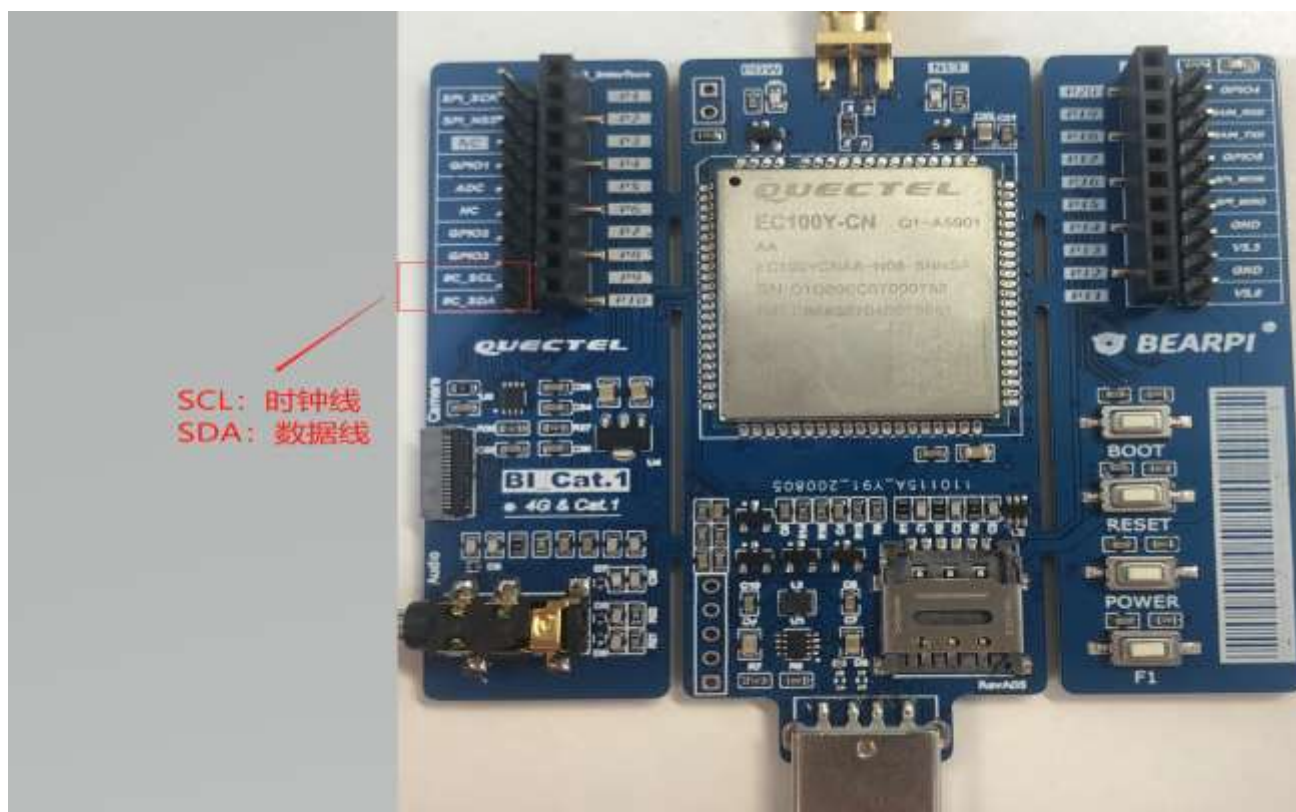


图 6: I2C 硬件连接

以下数据以 I2C 连接光照传感器为例。


```
< read i2c iRet=0, LIGHT=0x0256 498.333344>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x025c 503.333344>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x0290 546.666687>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x031e 665.000000>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x02ea 621.666687>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x02bb 582.500000>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x02d2 601.666687>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x02df 612.500000>
< BH1750_ENABLE >
< write i2c iRet=0 >
< write i2c iRet=0 >
< read i2c iRet=0, LIGHT=0x02e7 619.166687>
```

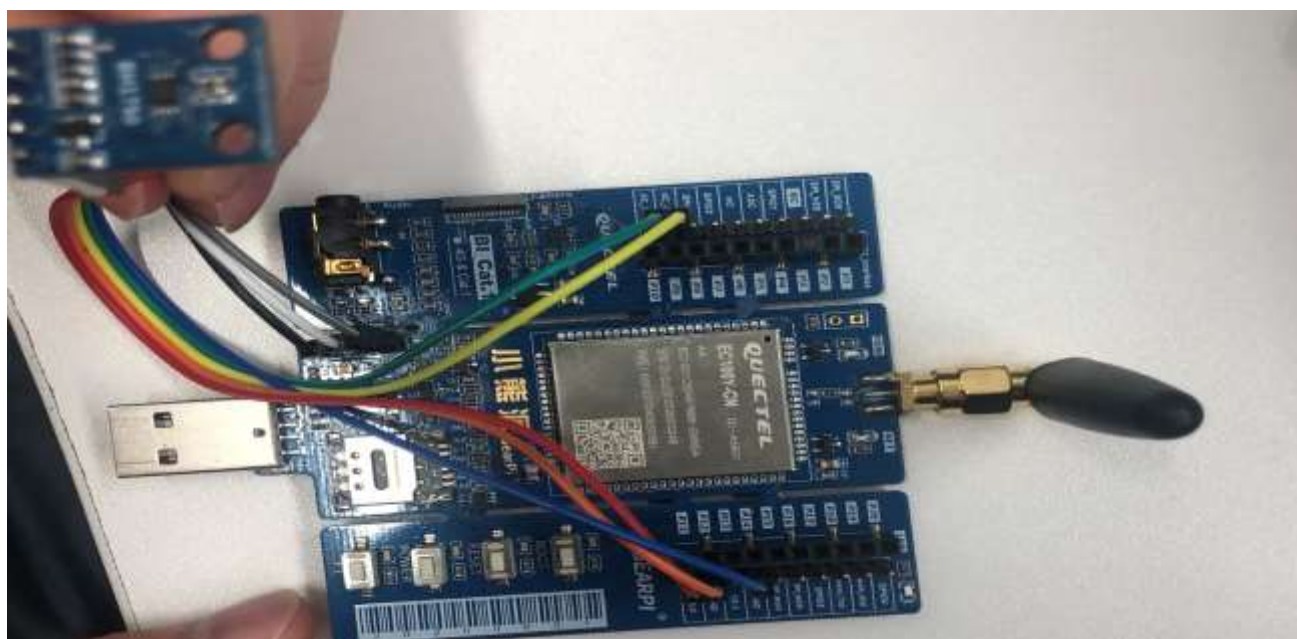


图 7: I2C 连接光照传感器

3 附录 A 术语缩写

表 1：术语缩写

缩写	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
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
GND	Ground	地
IC	Integrated Circuit	集成电路
I2C	Inter-Integrated Circuit	双向二线制同步串行总线
LCD	Liquid Crystal Display	液晶显示屏
MCU	Microprogrammed Control Unit	微程序控制器
MISO	Master In Slave Out	主机输入从机输出
MOSI	Master Out Slave In	主机输出从机输入
RX	Receive	接收
TX	Transmit	发送
SCK	Serial Clock	时钟信号线
SCL	Serial Clock Line	串行时钟线
SDA	Serial Data Line	串行数据线
SPI	Serial Peripheral Interface	串行外设接口
SS	Slave Select	片选信号线