QuecPython

GPIO 及 EXTI 使用说明

LTE 系列

版本：GPIO 及 EXTI 使用说明_V1.0

日期：2020-12-23

状态：临时文件

www.quectel.com

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

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2020-12-23	叶智强	初始版本

目录

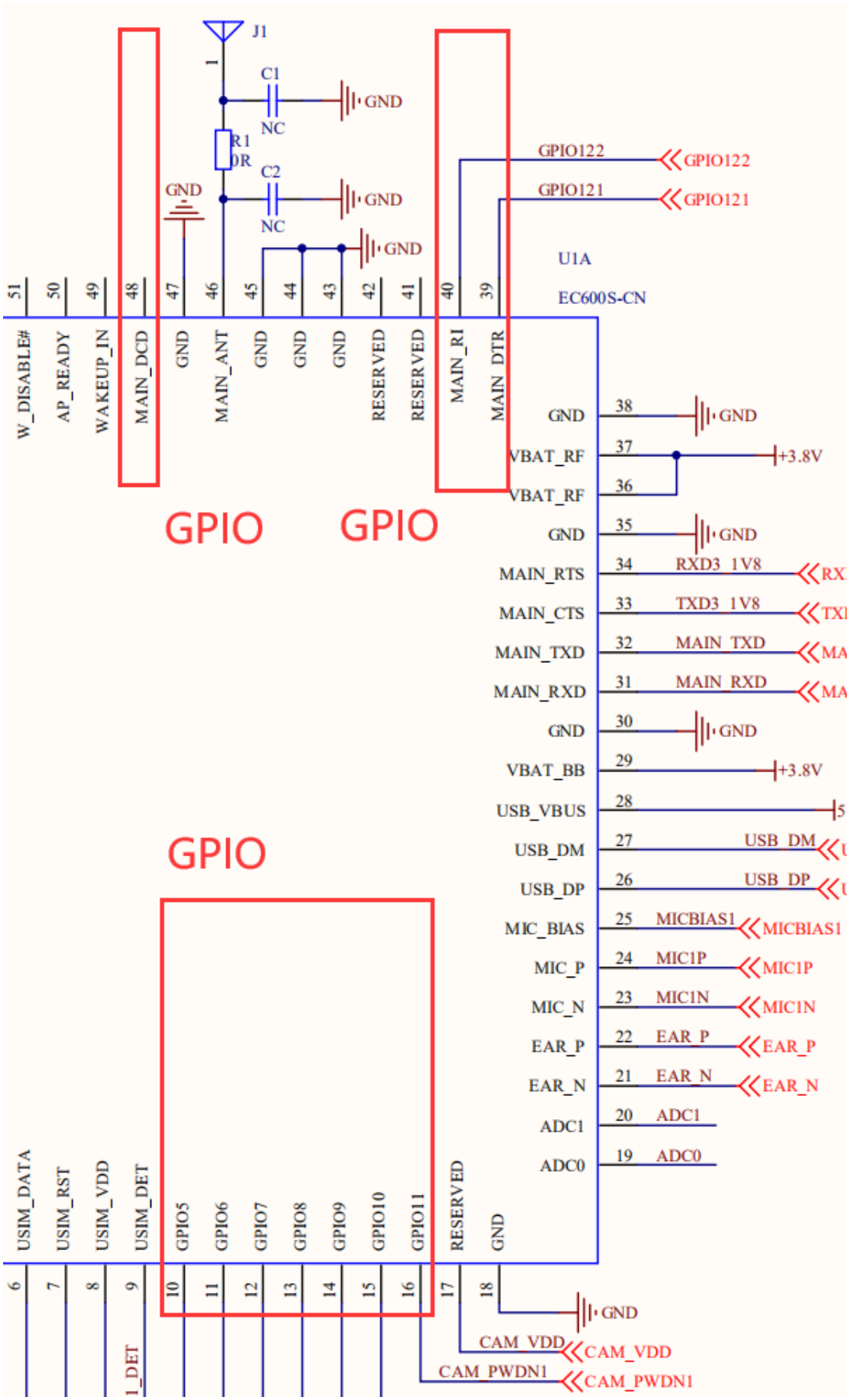
一、EC600S GPIO 基本概述

文档主要基于 EC600S 介绍如何使用 QuecPython_GPIO，GPIO 通常用于连接驱动 LED、蜂鸣器、继电器等等，同样也可以用来读取 KEY、开关状态、外部 IC 的引脚电平状态等等。通过本文你将了解到 GPIO 的所有设置参数及使用方法。

1. 硬件描述

目前开放共 10 个 GPIO，其他的 GPIO 逐渐开放，各个 GPIO 口连接位置如下表所示：

GPIO1 – 模组的 Pin10
GPIO2 – 模组的 Pin11
GPIO3 – 模组的 Pin12
GPIO4 – 模组的 Pin13
GPIO5 – 模组的 Pin14
GPIO6 – 模组的 Pin15
GPIO7 – 模组的 Pin16
GPIO8 – 模组的 Pin39
GPIO9 – 模组的 Pin40
GPIO10 – 模组的 Pin48


J1
C1
NC
GND
R1
0R
C2
NC
GND
GND
GND
GPIO122
GPIO121
GPIO122
GPIO121
U1A
EC600S-CN
GPIO
GPIO
GPIO
W_DISABLE#
AP_READY
WAKEUP_IN
MAIN_DCD
GND
MAIN_ANT
GND
GND
GND
RESERVED
RESERVED
MAIN_RI
MAIN_DTR
GND
VBAT_RF
VBAT_RF
GND
MAIN_RTS
MAIN_CTS
MAIN_TXD
MAIN_RXD
GND
VBAT_BB
USB_VBUS
USB_DM
USB_DP
MIC_BIAS
MIC_P
MIC_N
EAR_P
EAR_N
ADC1
ADC0
RXD3_1V8
TXD3_1V8
MAIN_TXD
MAIN_RXD
MICBIAS1
MIC1P
MIC1N
EAR_P
EAR_N
ADC1
ADC0
USIM_DATA
USIM_RST
USIM_VDD
USIM_DET
GPIO5
GPIO6
GPIO7
GPIO8
GPIO9
GPIO10
GPIO11
RESERVED
GND
CAM_VDD
CAM_PWDN1
CAM_VDD
CAM_PWDN1

2. 软件设计

2.1. 函数原型 Pin(GPIOn, direction, pullMode, level)，返回一个 pgio 对象，用于操作读写 IO 状态。
创建 gpio 对象：

gpio = Pin(GPIOn, direction, pullMode, level)

参数	类型	说明
GPIOn	int	GPIO1 – 模组的 Pin10 GPIO2 – 模组的 Pin11 GPIO3 – 模组的 Pin12 GPIO4 – 模组的 Pin13 GPIO5 – 模组的 Pin14 GPIO6 – 模组的 Pin15 GPIO7 – 模组的 Pin16 GPIO8 – 模组的 Pin39 GPIO9 – 模组的 Pin40 GPIO10 – 模组的 Pin48
direction	int	IN – 输入模式，OUT – 输出模式
pullMode	int	PULL_DISABLE – 浮空模式 PULL_PU – 上拉模式 PULL_PD – 下拉模式
level	int	0 - 设置引脚为低电平, 1- 设置引脚为高电平

2.2. 函数原型 read()，返回对应创建 gpio 对象的 Pin 脚输入电平状态。

获取 Pin 脚电平：

state= gpio.read()

参数	类型	说明
state	int	0 低电平，1 高电平

2.3. 函数原型 write(value)，设置对应创建 gpio 对象的 Pin 脚输出电平状态，返回设置结果。

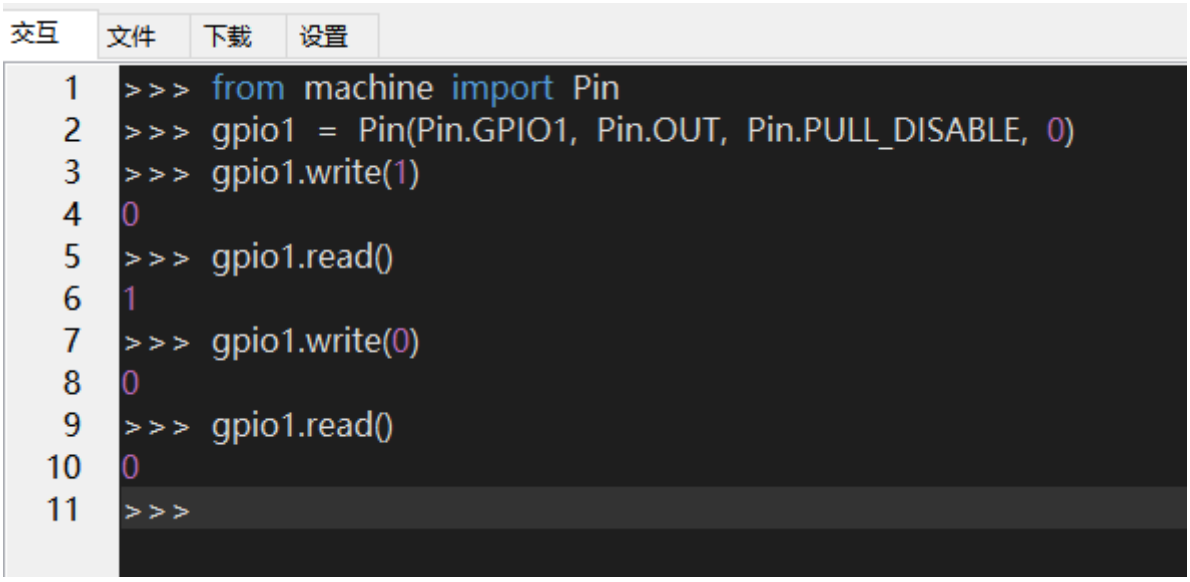
设置 Pin 脚电平：

ret = gpio.Pin.write(value)

参数	类型	说明
value	int	0 -当 PIN 脚为输出模式时，设置当前 Pin 脚输出低 1 -当 PIN 脚为输出模式时，设置当前 Pin 脚输出高
ret	int	0 成功 -1 失败

3. 交互操作

3.1. 使用 QPYcom 工具和模组进行交互，示例如下：



注意：

1. import Pin 即为让 Pin 模块在当前空间可见。
2. 只有 import Pin 模块，才能使用 Pin 内的函数和变量。

4. 下载验证

4.1. 下载.py 文件到模组运行，代码如下：

```
from machine import Pin


gpio1 = Pin(Pin.GPIO1, Pin.OUT, Pin.PULL_DISABLE, 0)
ret = gpio1.write(1)
print(ret)
ret = gpio1.read()
print(ret)
ret = gpio1.write(0)
print(ret)
ret = gpio1.read()
print(ret)
```

4.2. 配套 demo：

参考代码为文档同目录下的 GPIO.py 文件。

二、外部中断_基本概述

认识了普通的 GPIO 之后，接着介绍外部中断，普通 GPIO 可以随时查询 Pin 引脚的电平状态，但却不能及时发现电平的变化，而外部中断就能完美的解决这个问题。若设定了上升沿触发外部中断时，当电平从低电平上升到高电平瞬间，就会触发外部中断，从而在电平变化时立马执行回调函数。

5. 软件设计

5.1. 函数原型 ExtInt(GPIOn, mode, pull, callback)，返回一个 extint 对象，用于使能、禁止中断响应。
创建 extint 对象：

extint = ExtInt(GPIOn, mode, pull, callback)

参数	类型	说明
GPIOn	int	GPIO1 – 模组的 Pin10 GPIO2 – 模组的 Pin11 GPIO3 – 模组的 Pin12 GPIO4 – 模组的 Pin13 GPIO5 – 模组的 Pin14 GPIO6 – 模组的 Pin15 GPIO7 – 模组的 Pin16 GPIO8 – 模组的 Pin39 GPIO9 – 模组的 Pin40 GPIO10 – 模组的 Pin48
mode	int	IRQ_RISING – 上升沿触发 IRQ_FALLING – 下降沿触发 IRQ_RISING_FALLING – 上升和下降沿触发
pull	int	PULL_DISABLE – 浮空模式 PULL_PU – 上拉模式 PULL_PD – 下拉模式
callback	function	中断触发回调函数

5.2. 函数原型 enable()，设置对应创建 extint 对象的中断响应，返回设置结果。
允许中断

ret = extint.enable()

参数	类型	说明
ret	int	0 成功 -1 失败

5.3. 函数原型 disable()，设置对应创建 extint 对象的中断响应，返回设置结果。
禁止中断

ret = extint.disable()

参数	类型	说明
ret	int	0 成功 -1 失败

6. 交互操作

6.1. 使用 QPYcom 工具和模组进行交互，示例如下：

```
>>> from machine import ExtInt
>>> def fun(args):
...     print(args)
...     print('123456')
...
...
...
>>> extint1 = ExtInt(ExtInt.GPIO1, ExtInt.IRQ_FALLING, ExtInt.PULL_PU, fun)
>>> extint2 = ExtInt(ExtInt.GPIO2, ExtInt.IRQ_FALLING, ExtInt.PULL_PU, fun)
>>> extint1.enable()
0
>>> extint2.enable()
0
>>>
72
123456

71
123456
```

注意：回调函数 fun(args)中的 args，是引脚中断后返回的内部 GPIO 行号，一般用不上，但也要设置。

7. 下载验证

7.1. 下载.py 文件到模组运行，代码如下：

```
from machine import ExtInt
import utime


def fun1(args):
    print(args)
    print("1111111111111111111111")


def fun2(args):
    print(args)
    print("22222222222222222222")


extint1 = ExtInt(ExtInt.GPIO1, ExtInt.IRQ_FALLING, ExtInt.PULL_PU, fun1)
extint1.enable()

extint2 = ExtInt(ExtInt.GPIO2, ExtInt.IRQ_FALLING, ExtInt.PULL_PU, fun2)
extint2.enable()

while True:
    utime.sleep_ms(200)
    print("。。。。。。。。。")
```

注意：回调函数 fun(args)中的 args，是引脚中断后返回的内部 GPIO 行号，一般用不上，但必须设置。

7.2. 配套 demo：

参考代码为文档同目录下的 ExtInt.py 文件。

8.配套 demo：

参考代码的.py 文件在文档同文件夹下可找到。

9.专业名词解释：

低电平：通常用 0 来表示低电平
高电平：通常用 1 来表示高电平
上升沿：从低电平上升到高电平的边沿
下降沿：从高电平上升到低电平的边沿
回调函数：一个普通函数，在满足设定条件下被触发执行这个函数
浮空：Pin 引脚直出，没有默认电平，处于不稳定状态
上拉：Pin 引脚内部有个电阻拉到 VCC，默认为高电平
下拉：Pin 引脚内部有个电阻拉到 GND，默认为低电平
输入：Pin 引脚的电平状态随外部变化
输出：Pin 引脚的电平驱动外围电路
中断：停止执行当前的程序去执行另一段程序，这个过程叫中断