QuecPython

按键输入小实验

LTE 系列

版本：Quectel QuecPython 按键输入小实验 _V1.0

日期：2020-12-24

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

文档历史

修订记录

版本	日期	作者	变更表述
0.1	2020-12-24	周成柱	初始版本

目录

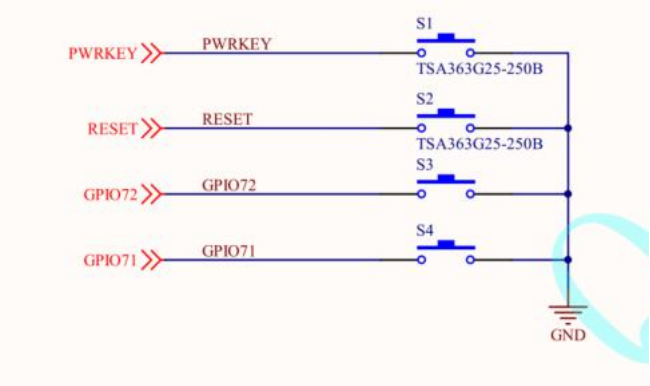
文档历史..... 2
1. 基本概述..... 4
1.1. 硬件资源..... 4
2. 实验设计..... 4
3. 代码实现..... 4
4. 实验操作..... 8
5. 专业名词..... 错误！未定义书签。
6. 参考文献..... 11

1.基本概述

本片文章主要简介EC600S ADC 硬件资源， 介绍quecpython ADC API，以及使用ADC 来检测当前光敏电阻的阻值。

1.1.硬件资源

EC600 引出了四个按键接口。参考 EC600S_QuecPython_EVB_V1.0_SCH.pdf 文档。



按键	引脚
S1	-
S2	-
S3	GPIO72
S4	GPIO71

当按键按下的时候， 我们可以检测到对应的引脚由 1 变为 0.

2.实验设计

代码一直轮询检测引脚状态。 分别检测两个按键，单击，双击，以及长按。

3.代码实现

```
'''
File: buttonclass.py
Project: button
File Created: Thursday, 24th December 2020 5:52:44 pm
Author: chengzhu.zhou
```

```
-----
Last Modified: Friday, 25th December 2020 5:30:48 pm
Modified By: chengzhu.zhou
-----
Copyright 2020 - 2020 quectel
'''


# copy from https://blog.csdn.net/qq997758497/article/details/80606710
from machine import Pin
import _thread
import utime as time


def Processing_button_fun(Alias, actionKey):
    print("{0} has {1} action".format(Alias, actionKey))
    pass


class ButtonClass():
    Alias = None
    Gpio_obj = None
    # check quiet click and long
    Status2 = None
    Status1 = None
    callbackfun = None
    timercount2 = None
    timercount1 = None
    # macro
    # readonly
    KEY_ACTION_QUIET = "quiet"
    KEY_ACTION_CLICK = "click"
    KEY_ACTION_DOUBLE = "double"
    KEY_ACTION_LONG = "long"
    # KEY_ACTION = {"quiet": "No key is generated ", "click": "Single key generation
",
    #               "double": "Double click the event", "long": "There are long pres
s events"}

    # KeyValue
    KEY_VALUE = {"press": 0, "normal": 1}

    #
    KEY_LONG_MAX_TIME = 2000
```

```
    def SetButtonStatu2(self, status="state0", time=0x00):
        self.Status2 = status
        self.timercount2 = time

    def SetButtonStatu1(self, status="state0", time=0x00):
        self.Status1 = status
        self.timercount1 = time

    def __nextstatus(self):
        pass

    def init(self, pin, callbackfun, Alias="gpio0"):
        self.Gpio_obj = Pin(pin, Pin.IN, Pin.PULL_DISABLE, 0)
        self.Alias = Alias
        self.callbackfun = callbackfun
        self.SetButtonStatu2("state0", 0x0)
        self.SetButtonStatu1("state0", 0x0)
        pass

        # return quiet click and long
    def __button_read_key(self):
        _Status = self.Status2
        keyValue = self.Gpio_obj.read()
        if _Status == "state0":
            if keyValue == self.KEY_VALUE["press"]:
                self.SetButtonStatu2("state1", 0x00)
            return self.KEY_ACTION_QUIET

            # Software chattering elimination
        if _Status == "state1":
            if keyValue == self.KEY_VALUE["press"]:
                self.SetButtonStatu2("state2", time.ticks_ms())
            else:
                # reset status
                self.SetButtonStatu2("state0", 0x00)
            return self.KEY_ACTION_QUIET
        elif _Status == "state2":
            if keyValue == self.KEY_VALUE["normal"]:
                # has click occur
                self.SetButtonStatu2("state0", self.timercount2)
                return self.KEY_ACTION_CLICK
            else:
                difftime = time.ticks_ms() - self.timercount2
                if difftime > self.KEY_LONG_MAX_TIME:
```

```
                    self.SetButtonStatu2("state0", 0x00)
                    return self.KEY_ACTION_LONG
        elif _Status == "state3":
            # Wait for the key to release
            if keyValue == self.KEY_VALUE["normal"]:
                self.SetButtonStatu2("state0", 0x00)
        return self.KEY_ACTION_QUIET

    def polling(self):
        # check has double click
        _Status = self.Status1
        _KeyStatus = self.__button_read_key()
        if _Status == "state0":
            if _KeyStatus != self.KEY_ACTION_CLICK:
                return _KeyStatus
            else:
                self.SetButtonStatu1("state1", time.ticks_ms())
                # No report
                return self.KEY_ACTION_QUIET
        elif _Status == "state1":
            difftime = time.ticks_ms() - self.timercount1
            if _KeyStatus == self.KEY_ACTION_CLICK:
                # Second time detected in a short time
                self.SetButtonStatu1("state0", 0x00)
                return self.KEY_ACTION_DOUBLE
            elif difftime >= 500:
                self.SetButtonStatu1("state0", 0x00)
                return self.KEY_ACTION_CLICK
        return _KeyStatus


#
def button_polling_thread(delay, PinList):
    ButtonList = []
    i = 0
    # init button
    for _pin in PinList:
        _temp = ButtonClass()
        _temp.init(_pin, Processing_button_fun, "button{0}".format(i))
        ButtonList.append(_temp)
        i = i + 1
    # Polling button
    i = 10
    while i:
        for button in ButtonList:
```

```
            action = button.polling()
            if action != ButtonClass.KEY_ACTION_QUIET:
                # has press
                button.callbackfun(button.Alias, action)
                i = i - 1
        time.sleep_ms(10)
    print("button thread has exited")


if __name__ == "__main__":
    # creat a thread Check key status
    _thread.start_new_thread(button_polling_thread,
                             (1, [Pin.GPIO1, Pin.GPIO2]))
```

4.实验操作

1. 将 buttonclass.py 烧录到/usr 目录下。
2. 使用下面的命令执行脚本。

```
>>> uos.getcwd()
'/'
>>> uos.listdir()
['usr', 'bak']
>>> uos.chdir('usr')
>>> uos.listdir()
['apn_cfg.json', 'maonv.mp3', 'test.py', 'buttonclass.py']
>>> import example
>>> example.exec('usr/buttonclass.py')
```

3. 按下板卡按键，查看打印日志。

最终我们可以在串口看到，阻值输出的变化如下。

```
button0 has double action

button0 has click action

button0 has click action

button0 has long action

button0 has long action
```

```
button0 has click action

button1 has click action

button1 has double action

button0 has double action

button0 has double action
```

5.其他实验引用 buttonclass 脚本

5.1.测试代码

```
'''
File: test_buttonclass.py
Project: button
File Created: Friday, 25th December 2020 5:42:17 pm
Author: chengzhu.zhou
-----
Last Modified: Friday, 25th December 2020 5:42:41 pm
Modified By: chengzhu.zhou
-----
Copyright 2020 - 2020 quectel
'''
import utime as time
from buttonclass import ButtonClass
from machine import Pin
import _thread


def Processing_button_fun(Alias, actionKey):
    if actionKey == ButtonClass.KEY_ACTION_CLICK:
        print("test: has click trigger")
        pass
    elif actionKey == ButtonClass.KEY_ACTION_DOUBLE:
        print("test: has double click trigger")
        pass
    elif actionKey == ButtonClass.KEY_ACTION_LONG:
        print("test: has long click trigger")
```

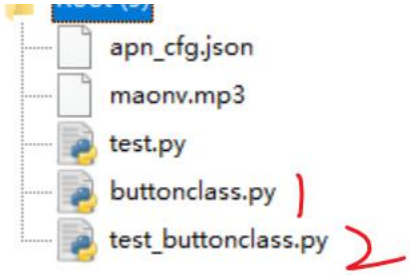
```
        pass
    pass


def button_polling_thread(delay, PinList):
    ButtonList = []
    i = 0
    # init button
    for _pin in PinList:
        _temp = ButtonClass()
        _temp.init(_pin, Processing_button_fun, "button{0}".format(i))
        ButtonList.append(_temp)
        i = i + 1
    # Polling button
    i = 10
    while i:
        for button in ButtonList:
            action = button.polling()
            if action != ButtonClass.KEY_ACTION_QUIET:
                # has press
                button.callbackfun(button.Alias, action)
                i = i - 1
        time.sleep_ms(10)
    print("button thread has exited")


if __name__ == "__main__":
    # creat a thread Check key status
    _thread.start_new_thread(button_polling_thread,
                             (1, [Pin.GPIO1, Pin.GPIO2]))
```

5.2.测试步骤

1. 首先将 test_buttonclass.py 将 烧录到和 buttonclass.py 同级目录。



2. 然后使用命令行的方式切换到对应目录。我这里是/usr 目录。

```
>>> uos.getcwd()
'/'
>>> uos.listdir()
['usr', 'bak']
>>> uos.chdir('usr')
>>> uos.listdir()
['apn_cfg.json', 'maonv.mp3', 'test.py', 'buttonclass.py', 'test_buttonclass.py']
>>> import example
>>> example.exec('usr/test_buttonclass.py')
```

运行脚本以后，按下按键即可。下面是测试 log

```
test: has click trigger

test: has click trigger

test: has double click trigger

test: has double click trigger

test: has click trigger

test: has long click trigger
test: has long click trigger

test: has long click trigger

test: has long click trigger

test: has click trigger
button thread has exited
```

6.参考文献

[1]. EC600S_QuecPython_EVB_V1.0_SCH.pdf
[2]. 光敏电阻特性