浙江大学 实验报告

课程名称：嵌入式系统　　指导老师：蔡铭　　学生姓名：李磊

实验名称：夏任务 128：数码管骰子　　实验类型：操作实践　　学生学号：3110102782

一、实验目的和要求

实验目的：

使用 pcDuino 实现一个数码管筛子，数字筛子的基本要求是数码管（只需要一个数字）快速在 1-8 之间循环跳动，当按下按键时，数码管定格在当前的数字上，不再跳动。

实验要求：

实验报告要包括 Fritzing 电路图和源代码

二、实验内容和原理

使用 pcDuino 实现一个数码管筛子，数字筛子的基本要求是数码管（只需要一个数字）快速在 1-8 之间循环跳动，当按下按键时，数码管定格在当前的数字上，不再跳动。

三、主要仪器设备

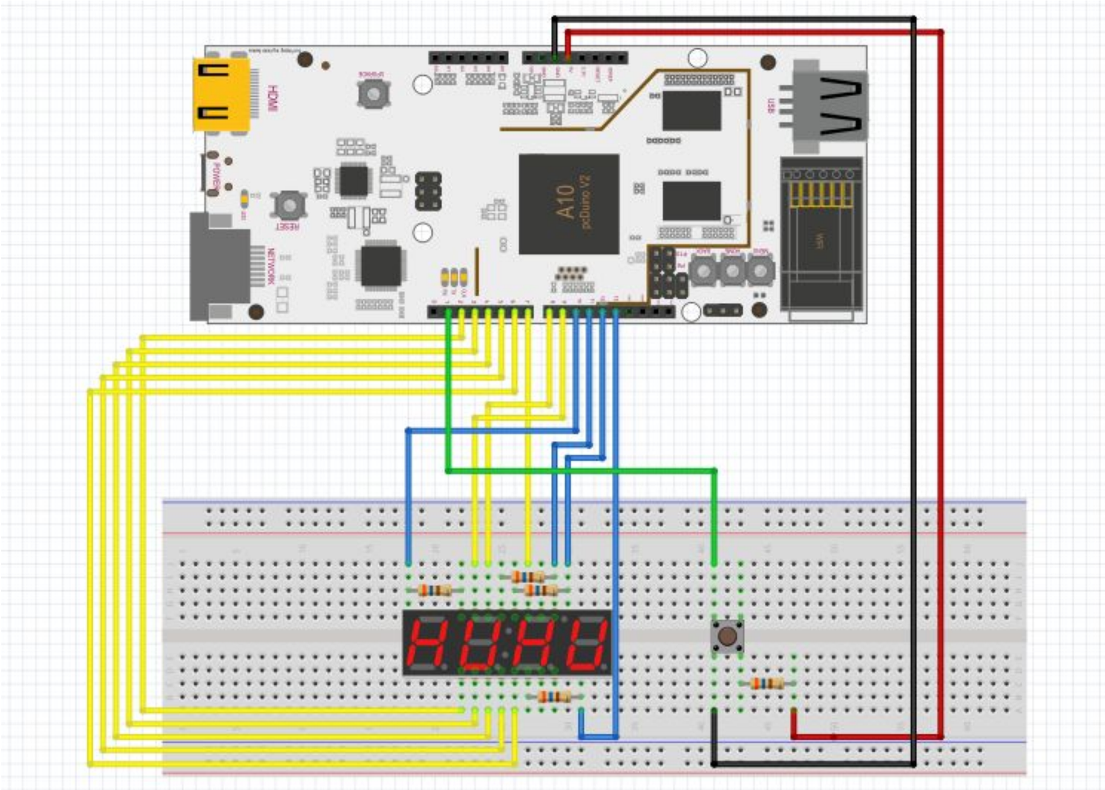
1. acaDia 板一块；
2. 5V/2A 电源一个；
3. microUSB 线一根；
4. 4 位 7 段数码管；
5. USB-TTL 串口线一根（FT232RL 芯片或 PL2303 芯片）；
6. 按钮若干；
7. 电阻若干；
8. 公线若干；
9. PC 一台；
10. 以太网线一根。

四、操作方法和实验步骤

1. 连接硬件外围设备；

面包板 X1、4 位 7 段数码管 X1、360ohm 电阻 X5、按钮 X1、面包线若干，fritzing 图如下所示。

由于 fritzing 找不到 acaDia 这块开发板，故使用 pcduinoV2 代替：



在~/python-pcduino/pcduino 下建立 clock.py

python 源代码：

```
import gpio
import time
import random
cnt = 0
channel                                                                                       =
{0:'gpio2',1:'gpio3',2:'gpio4',3:'gpio5',4:'gpio6',5:'gpio7',6:'gpio8',7:'gpio9',8:'gpio10',9:'gpio11',1
0:'gpio12',11:'gpio13'}
sec=0
digit4=0
digit3=0
digit2=0
digit1=0
btn_read=0
on=0
HIGH = gpio.HIGH
LOW = gpio.LOW
INPUT = gpio.INPUT
OUTPUT = gpio.OUTPUT
```

```
def delay(ms):                          #delay()程序
                                         time.sleep(1.0*ms/1000)
def setup() :                           #初始化
                                         for i in range(0,8):
                                           gpio.pin_mode(channel[i],OUTPUT)
                                         for i in range(9,12):
                                           gpio.pin_mode(channel[i],OUTPUT)
                                         gpio.pin_mode('gpio1',INPUT)

def display_digit(digit):                #数码管显示数值
                                         if digit==0:
                                           gpio.digital_write(channel[0],LOW)
                                           gpio.digital_write(channel[1],LOW)
                                           gpio.digital_write(channel[2],HIGH)
                                           gpio.digital_write(channel[3],LOW)
                                           gpio.digital_write(channel[4],HIGH)
                                           gpio.digital_write(channel[5],LOW)
                                           gpio.digital_write(channel[6],LOW)
                                           gpio.digital_write(channel[7],LOW)

                                         elif digit== 1:
                                           gpio.digital_write(channel[0],HIGH)
                                           gpio.digital_write(channel[1],HIGH)
                                           gpio.digital_write(channel[2],HIGH)
                                           gpio.digital_write(channel[3],LOW)
                                           gpio.digital_write(channel[4],HIGH)
                                           gpio.digital_write(channel[5],LOW)
                                           gpio.digital_write(channel[6],HIGH)
                                           gpio.digital_write(channel[7],HIGH)

                                         elif digit== 2:
                                           gpio.digital_write(channel[0],LOW)
                                           gpio.digital_write(channel[1],LOW)
                                           gpio.digital_write(channel[2],HIGH)
                                           gpio.digital_write(channel[3],HIGH)
                                           gpio.digital_write(channel[4],LOW)
                                           gpio.digital_write(channel[5],LOW)
                                           gpio.digital_write(channel[6],HIGH)
                                           gpio.digital_write(channel[7],LOW)

                                         elif digit== 3:
                                           gpio.digital_write(channel[0],HIGH)
                                           gpio.digital_write(channel[1],LOW)
                                           gpio.digital_write(channel[2],HIGH)
                                           gpio.digital_write(channel[3],LOW)
                                           gpio.digital_write(channel[4],LOW)
                                           gpio.digital_write(channel[5],LOW)
                                           gpio.digital_write(channel[6],HIGH)
                                           gpio.digital_write(channel[7],LOW)

                                         elif digit== 4:
                                           gpio.digital_write(channel[0],HIGH)
```

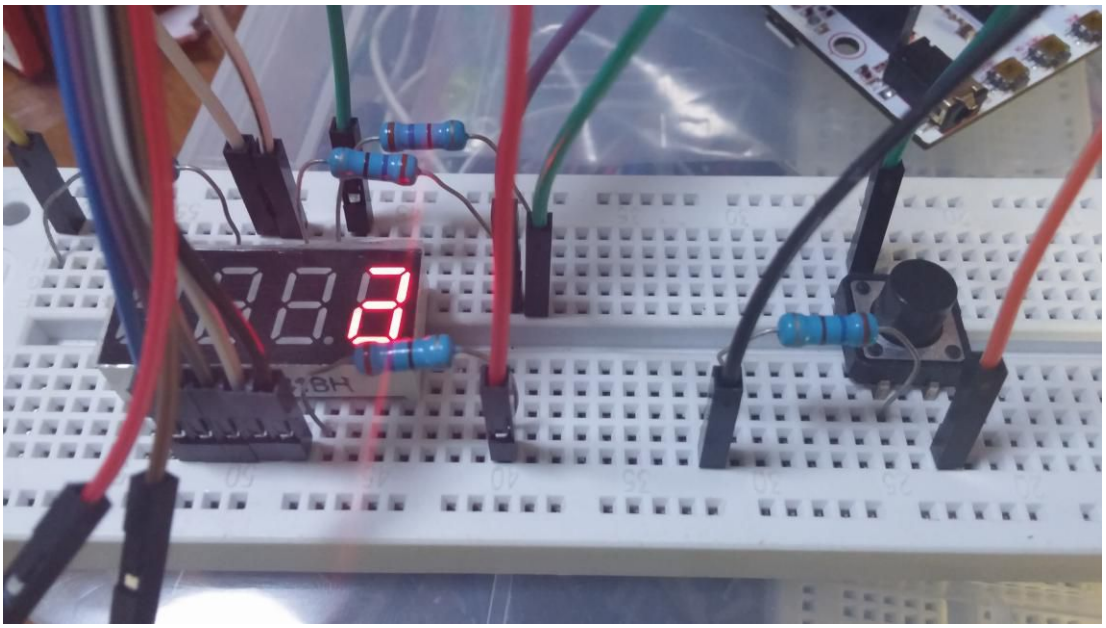
```
    gpio.digital_write(channel[1],HIGH)
    gpio.digital_write(channel[2],HIGH)
    gpio.digital_write(channel[3],LOW)
    gpio.digital_write(channel[4],LOW)
    gpio.digital_write(channel[5],LOW)
    gpio.digital_write(channel[6],LOW)
    gpio.digital_write(channel[7],HIGH)

  elif digit== 5:
    gpio.digital_write(channel[0],HIGH)
    gpio.digital_write(channel[1],LOW)
    gpio.digital_write(channel[2],HIGH)
    gpio.digital_write(channel[3],LOW)
    gpio.digital_write(channel[4],LOW)
    gpio.digital_write(channel[5],HIGH)
    gpio.digital_write(channel[6],LOW)
    gpio.digital_write(channel[7],LOW)

  elif digit== 6:
    gpio.digital_write(channel[0],LOW)
    gpio.digital_write(channel[1],LOW)
    gpio.digital_write(channel[2],HIGH)
    gpio.digital_write(channel[3],LOW)
    gpio.digital_write(channel[4],LOW)
    gpio.digital_write(channel[5],HIGH)
    gpio.digital_write(channel[6],LOW)
    gpio.digital_write(channel[7],LOW)

  elif digit== 7:
    gpio.digital_write(channel[0],HIGH)
    gpio.digital_write(channel[1],HIGH)
    gpio.digital_write(channel[2],HIGH)
    gpio.digital_write(channel[3],LOW)
    gpio.digital_write(channel[4],HIGH)
    gpio.digital_write(channel[5],LOW)
    gpio.digital_write(channel[6],HIGH)
    gpio.digital_write(channel[7],LOW)

  elif digit== 8:
    gpio.digital_write(channel[0],LOW)
    gpio.digital_write(channel[1],LOW)
    gpio.digital_write(channel[2],HIGH)
    gpio.digital_write(channel[3],LOW)
    gpio.digital_write(channel[4],LOW)
    gpio.digital_write(channel[5],LOW)
    gpio.digital_write(channel[6],LOW)
    gpio.digital_write(channel[7],LOW)

  elif digit== 9:
    gpio.digital_write(channel[0],HIGH)
    gpio.digital_write(channel[1],LOW)
    gpio.digital_write(channel[2],HIGH)
    gpio.digital_write(channel[3],LOW)
    gpio.digital_write(channel[4],LOW)
```

```
                                    gpio.digital_write(channel[5],LOW)
                                    gpio.digital_write(channel[6],LOW)
                                    gpio.digital_write(channel[7],LOW)

def get_digit():                    #获得数值
                                    global digit1,digit2,digit3,digit4
                                    digit1=random.randint(1,8)

def loop() :                        #根据按钮来决定是否暂停掷骰子
                                    global cnt
                                    global btn_read,sec,on


                                    gpio.digital_write(channel[8],LOW)
                                    gpio.digital_write(channel[9],LOW)
                                    gpio.digital_write(channel[10],LOW)
                                    gpio.digital_write(channel[11],HIGH)
                                    display_digit(digit1)
                                    delay(5)
                                    cnt=cnt+5

                                    btn_read=gpio.digital_read('gpio1')
                                    if btn_read==1:
                                      delay(20)
                                      btn_read=gpio.digital_read('gpio1')
                                      if btn_read ==1:
                                            if on==1:
                                                  print("pause")
                                                  on=0
                                      else :
                                            print("start")
                                            on=1


                                    if cnt>=100 and on==1 :
                                      cnt=0
                                      get_digit()
def main():
                                    setup()
                                    while(1):
                                      loop()
main()
```

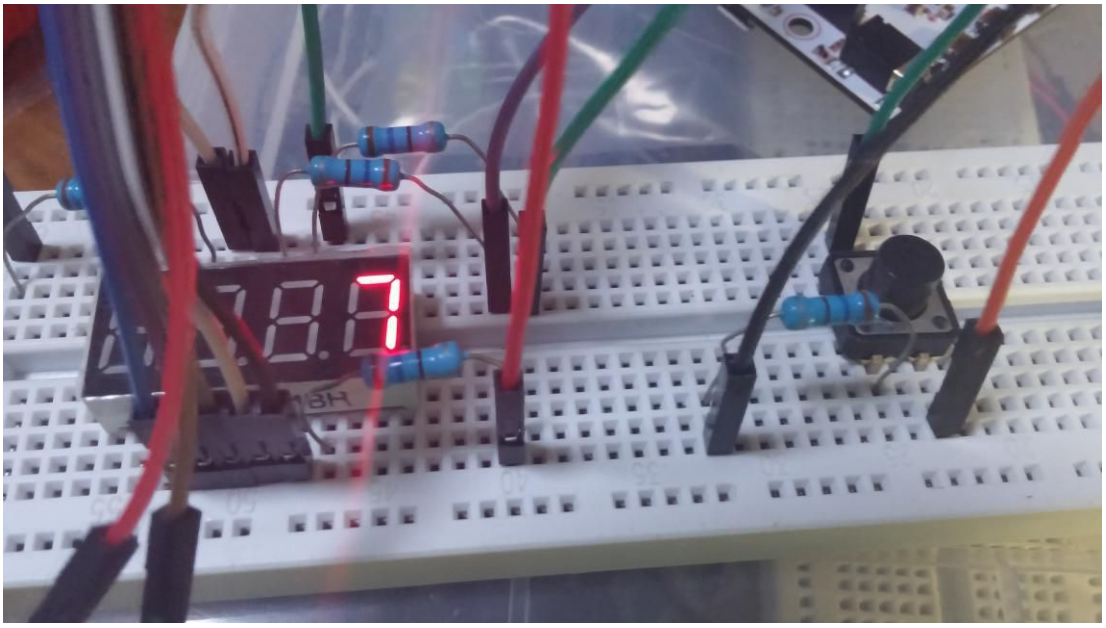
2. 将代码下入板子进行验证

```
$ python ./clock.py
```

起初，7 段数码管显示 0，掷骰子未开始。此时按下按钮，console 输出“start”，开始掷骰子，中途拍照，LED 显示数值：



再次按下按钮，console 输出“pause”，计数停止，显示数值 7：



再次按下按钮，console 输出“start”，计数再次开始。。。。。。

Console 输出结果如下：

```
root@Acadia:~/Desktop/python-pcduino-master/pcduino# python ./clock.py
start
pause
start
pause
start
pause
start
pause
start
pause
```

五、实验数据记录和处理

使用 pcDuino 实现一个数码管筛子，数字筛子的基本要求是数码管（只需要一个数字）快速在 1-8 之间循环跳动，当按下按键时，数码管定格在当前的数字上，不再跳动。

成功。

视频地址：http://v.youku.com/v_show/id_XOTYwNDI0Mjg0.html

六、实验结果与分析

实验成功。

七、讨论、心得

本次实验我抛弃了 Arduino IDE，继续使用 python 进行实验，调用 gpio 接口，并学会了使用七段数码管。