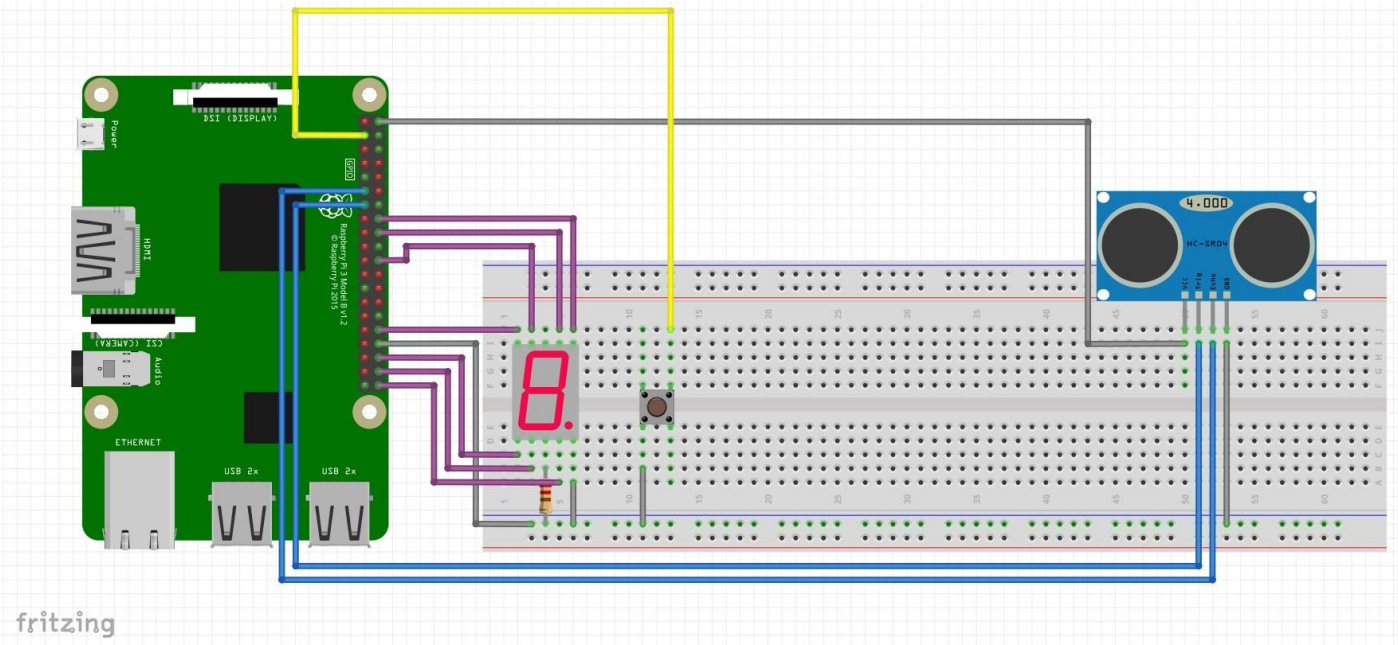
實驗二: Key,7-segment, and Ultrasonic

班級: 資科三甲

姓名: 謝妤婕

學號: 110816032

一、電路圖



找不到一樣大顆的按鈕圖例 QAQQQ 只好用小顆的代替。

二、程式

需求
(1) 7-segment 顯示"0"，Terminal 顯示 ultrasonic 偵測到障礙物的距離。 (2) Key pressed，7-segment 顯示"1"(0+1)，Terminal 顯示 ultrasonic 偵測到障礙物的距離 repeat (1)到(2)，7-segment 顯示累加的效果:0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, 0, 1, 2, ……
程式說明
步驟(1)對應函式 stepInit()，步驟(2)對應函式 main()中的函式 原需求可理解為按鈕按下後執行 「Terminal 顯示 ultrasonic 偵測到障礙物的距離」和「七解碼顯示[(計數器數值+1)%16]」 在 stepInit()先初始化所有數值和 pin 接著呼叫第一次的 ultrasonicDistance()計算距離 此後每次按下按鈕都會呼叫 keyAndSevenSegmentControl() 執行 ultrasonicDistance()計算距離和 sevenSegmentControl()更新七解碼狀態

程式

```
import RPi.GPIO as GPIO
import math
from time import time
from time import sleep

GPIO.setmode(GPIO.BCM)
pin_segA = 24
pin_segB = 23
pin_segC = 21
pin_segD = 20
pin_segE = 16
pin_segF = 25
pin_segG = 12
pin_ultraTrig = 27
pin_ultraEcho = 17
pin_key = 2
sevenSegmentPin = [
   pin_segG, pin_segF, pin_segE, pin_segD,
   pin_segC, pin_segB, pin_segA
]

sevenSegmentStringDict = {
   0: "0111111",
   1: "0000110",
   2: "1011011",
   3: "1001111",
   4: "1100110",
   5: "1101101",
   6: "1111101",
   7: "0000111",
   8: "1111111",
   9: "1101111",
   10: "1110111",
   11: "1111100",
   12: "0111001",
   13: "1011110",
   14: "1111001",
   15: "1110001"
}
```

```
userPressCount = 0

def stepInit():
  try:
    GPIO.setup(pin_key, GPIO.IN, GPIO.PUD_UP)
    GPIO.setup(pin_ultraEcho, GPIO.IN)
    GPIO.setup(pin_ultraTrig, GPIO.OUT)
    for pin in sevenSegmentPin:
      GPIO.setup(pin, GPIO.OUT)
    print("========== GPIO.setup done ==========")
    for pinIndex in range(7):
      value = int(sevenSegmentStringDict[0][pinIndex])
      GPIO.output(sevenSegmentPin[pinIndex], value)
    print("====== 7-Segment setup done =========")
    GPIO.output(pin_ultraTrig, False)
    print("====== ultraTrig setup done =========")
    global userPressCount
    userPressCount = 0
    print("====== variable setup done ==========")
    print("\n\n")
    print("====================================")
    print("====================================")
    print("==== press button to get distance ===")
    print("==== press keyboard ^C to finish ====")
    print("====================================")
    print("====================================")
    print("\n\n")
  except:
    print("now at state stepInit()")

def sevenSegmentControl(number):
  try:
    for pinIndex in range(7):
      value = int(sevenSegmentStringDict[number][pinIndex])
      GPIO.output(sevenSegmentPin[pinIndex], value)
  except:
    print("now at state sevenSegmentControl()")
    print("now number = ", number)
```

```
def ultrasonicEcho(value, timeout):
   count = timeout
   while GPIO.input(pin_ultraEcho) != value and count > 0:
      count = count - 1

def ultrasonicDistance():
   try:
      GPIO.output(pin_ultraTrig, True)
      sleep(0.001)
      GPIO.output(pin_ultraTrig, False)
      ultrasonicEcho(True, 5000)
      start = time()
      ultrasonicEcho(False, 5000)
      end = time()
      distance = round(((end-start) * 34000 / 2), 3)
      print("the distance = ", distance, " cm")
   except:
      print("now at state ultrasonicDistance()")

def keyAndSevenSegmentControl(pin_key):
   ultrasonicDistance()
   global userPressCount
   userPressCount += 1
   userPressCount %= 16
   sevenSegmentControl(userPressCount)

def main():
   stepInit()
   try:
      ultrasonicDistance()
      GPIO.add_event_detect(pin_key, GPIO.FALLING, callback=keyAndSevenSegmentControl, bouncetime=200)
      while (True):
         pass
   except KeyboardInterrupt:
      print("\n\n", "============= user STOP =============")
   except:
      print("\n\n", "========== other Exception ==========")
   finally:
      GPIO.cleanup()
      print("========== GPIO cleanup done ========")
      print("============ end of main() ==========")
```

```
main()
```

需求

(1) 7-Segment 由 Key 來控制
(2) Ultrasonic 不再被 Key 控制，而是獨立在主程式運作，每隔 3 秒顯示障礙物的距離

程式說明

原需求可理解為「Terminal 顯示 ultrasonic 偵測到障礙物的距離」和「按鈕計數器」
兩個功能同時獨立運作，也就是需要兩條 thread 0u0
keyAndSevenSegmentControl()僅留下計數、sevenSegmentControl()的功能
而 ultrasonicDistance()獨立於主程式中運作

以下原始碼僅展示改寫的部分 0u0
即 stepInit()、keyAndSevenSegmentControl()、main()
作業完整原始碼 ： https://colab.research.google.com/drive/1VnnKBSRmvi67gN_Qd4MrEBAymkH_Hn0T?usp=sharing

程式

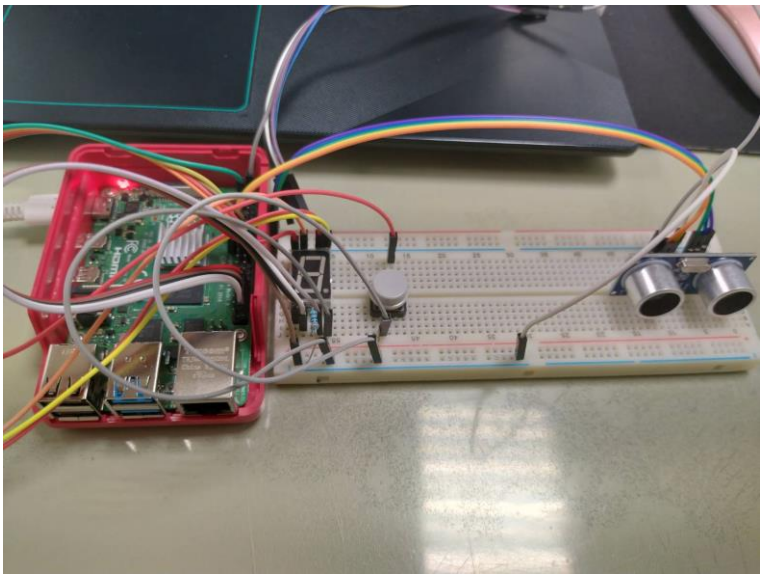
```
def stepInit():
  try:
    GPIO.setup(pin key, GPIO.IN, GPIO.PUD UP)
    GPIO.setup(pin ultraEcho, GPIO.IN)
    GPIO.setup(pin_ultraTrig, GPIO.OUT)
    for pin in sevenSegmentPin:
      GPIO.setup(pin, GPIO.OUT)
    print("========== GPIO.setup done ==========")
    for pinIndex in range(7):
      value = int(sevenSegmentStringDict[0][pinIndex])
      GPIO.output(sevenSegmentPin[pinIndex], value)
    print("====== 7-Segment setup done =========")
    GPIO.output(pin_ultraTrig, False)
    print("====== ultraTrig setup done =========")
    global userPressCount
    userPressCount = 0
    print("====== variable setup done ==========")
    print("\n\n")
    print("====================================")
    print("====================================")
    print("=== press button to ++ the number ===")
    print("==== press keyboard ^C to finish ====")
    print("====================================")
    print("====================================")
    print("\n\n")
  except:
    print("now at state stepInit()")
```

```
def keyAndSevenSegmentControl(pin key):
  # ultrasonicDistance()
  global userPressCount
  userPressCount += 1
  userPressCount %= 16
  sevenSegmentControl(userPressCount)

def main():
  stepInit()
  try:
    GPIO.add_event_detect(pin_key, GPIO.FALLING, callback=keyAndSevenSegmentControl, bouncetime=200)
    while (True):
      ultrasonicDistance()
      sleep(3)
   except KeyboardInterrupt:
    print("\n\n")
    print("============= user STOP =============")
  except:
    print("\n\n")
    print("========== other Exception ==========")
  finally:
    GPIO.cleanup()
    print("========== GPIO cleanup done ========")
    print("============ end of main() ==========")

main()
```

三、影片連結



YouTube 影片連結 ： https://youtu.be/f457qV9jYiA （原實驗 shell）

YouTube 影片連結 ： https://youtu.be/F0ysTouVhqo （原實驗麵包板）

分開控制的就沒錄了，想說影片也看不出來_(:3」∠)_

四、困難

不小心又一次過了，完全沒有卡關、(✿◠▽◠)ノ
硬要說遇到什麼困難，就是按鈕的腳太短，好難插到麵包板上、拔起來都用噴的。

五、心得

本來想說加一下家裡的網路，結果又回到 ping 不到板板的地獄了。
隔天看了一下路由器發現他是 5G 的，而且因為家裡的印表機要連線所以不能改 2.4G 試試，只好連回自己手機的網路了 QuQ
雖然不確定是不是頻率的問題，但我學會了遠距不要亂改網路設定，不然會嚇到寢食難安 QuQ