



# Χρήση Αισθητήρων

ΜΑΘΗΜΑΤΑ RASPBERRY

ΒΑΣΙΛΕΙΟΥ ΜΑΡΙΟΣ

ΜΑΝΟΣ ΝΙΚΟΛΑΟΣ

ΔΟΥΜΑ ΑΝΑΣΤΑΣΙΑ

ΚΑΒΑΛΛΙΕΡΑΤΟΥ ΕΡΓΙΝΑ

# Περιεχόμενα

- ▶ Αισθητήρας Passive Infrared PIR
- ▶ Διακόπτης ελέγχου – ρελέ
- ▶ Ποτενσιόμετρο
- ▶ Αισθητήρας ανίχνευσης ήχου
- ▶ Αισθητήρας κλίσης

# Αισθητήρας Passive Infrared (PIR)

3/25

Source: <https://projects.raspberrypi.org/en/projects/physical-computing/11>

- ▶ Ένας αισθητήρας PIR ανιχνεύει αλλαγές στην ποσότητα υπέρυθρης ακτινοβολίας που λαμβάνει. Όταν υπάρχει σημαντική αλλαγή στην ποσότητα υπέρυθρης ακτινοβολίας που ανιχνεύει, τότε στέλνεται ένα σήμα
- ▶ Αυτό σημαίνει ότι ένας αισθητήρας PIR μπορεί να ανιχνεύσει πότε ένας άνθρωπος (ή οποιοδήποτε ζώο) κινείται μπροστά του

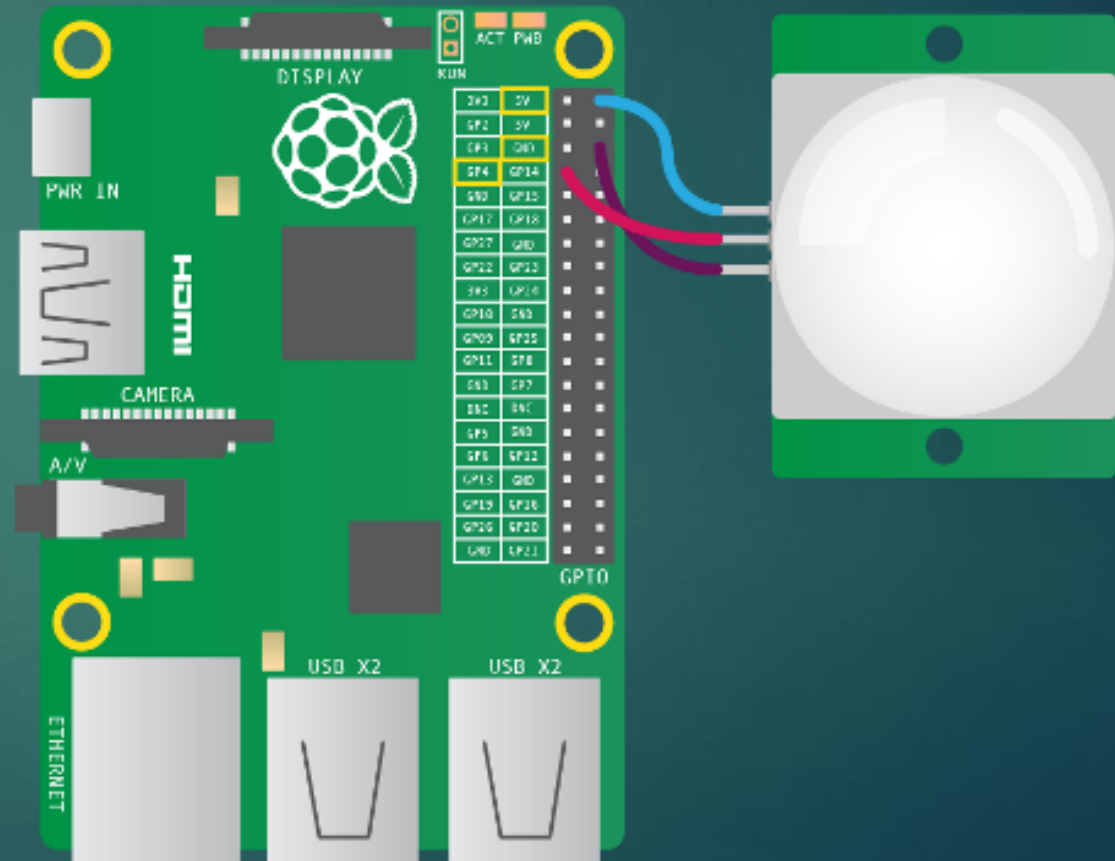


# Συνδεσμολογία PIR αισθητήρα (1/2)

4/25

Source: <https://projects.raspberrypi.org/en/projects/physical-computing/11>

- ▶ Το σήμα που εκπέμπεται όταν ο PIR ανιχνεύει κίνηση πρέπει να ενισχυθεί και έτσι πρέπει να τροφοδοτηθεί. Υπάρχουν τρεις ακροδέκτες στον PIR οι οποίες φέρουν την ένδειξη Vcc, Gnd και Out:

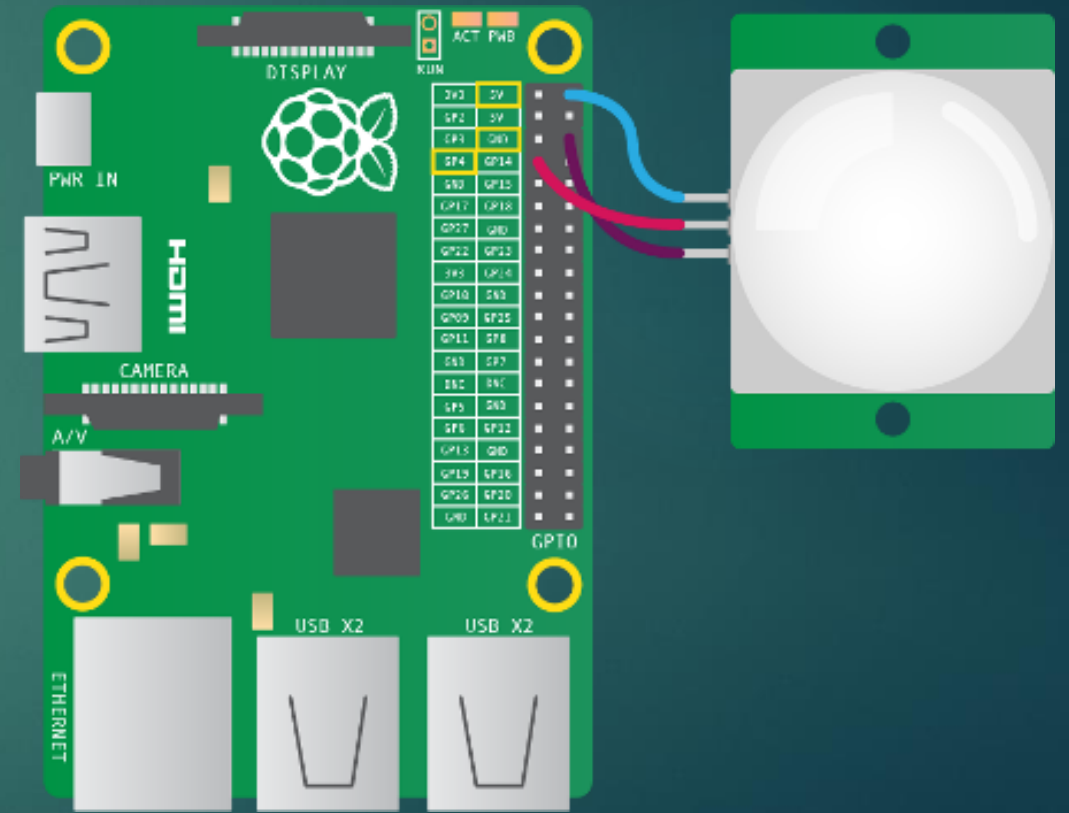


# Συνδεσμολογία PIR αισθητήρα (2/2)

5/25

Source: <https://projects.raspberrypi.org/en/projects/physical-computing/11>

- ▶ Ο ακροδέκτης Vcc του αισθητήρα πρέπει να συνδεθεί στον ακροδέκτη 5V του Raspberry Pi
- ▶ Ο ακροδέκτης Gnd του αισθητήρα πρέπει να συνδεθεί στον ακροδέκτη Gnd του Raspberry Pi
- ▶ Ο ακροδέκτης Out του αισθητήρα μπορεί να συνδεθεί σε οποιοδήποτε GPIO ακροδέκτη του Raspberry Pi

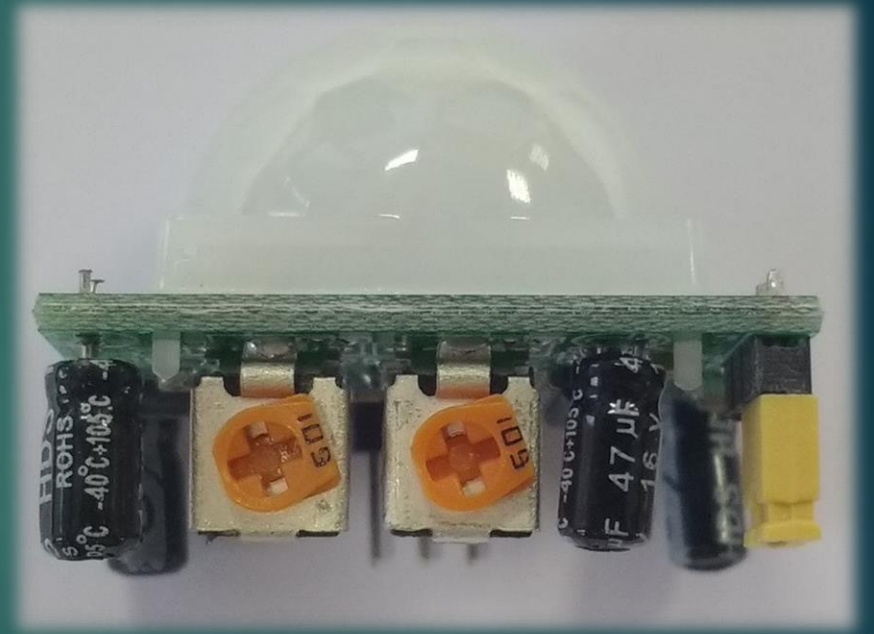


# Ρύθμιση PIR αισθητήρα (1/2)

6/25

Source: <https://projects.raspberrypi.org/en/projects/physical-computing/11>

- ▶ Οι περισσότεροι αισθητήρες PIR έχουν δύο ποτενσιόμετρα ενσωματωμένα στην πλακέτα τους. Αυτά μπορούν να ελέγξουν την ευαισθησία των αισθητήρων, καθώς και το χρονικό διάστημα για το οποίο θα σηματοδοτεί το PIR όταν ανιχνεύεται κίνηση
- ▶ Στην διπλανή εικόνα, το ποτενσιόμετρο στα δεξιά ελέγχει την ευαισθησία και το ποτενσιόμετρο στα αριστερά ελέγχει το χρονικό όριο. Εδώ, και οι δύο στρέφονται πλήρως αριστερόστροφα, πράγμα που σημαίνει ότι η ευαισθησία και το χρονικό όριο είναι στο χαμηλότερο επίπεδο

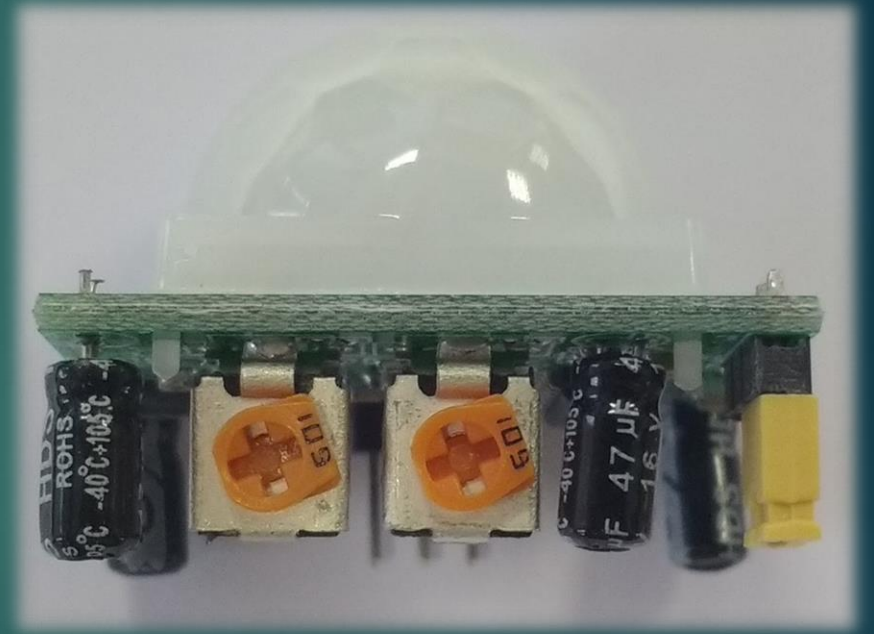


# Ρύθμιση PIR αισθητήρα (2/2)

7/25

Source: <https://projects.raspberrypi.org/en/projects/physical-computing/11>

- ▶ Όταν το χρονικό όριο στρέφεται πλήρως αριστερόστροφα, ο PIR θα βγάλει ένα σήμα για περίπου 2,5 δευτερόλεπτα, κάθε φορά που ανιχνεύεται κίνηση. Εάν το ποτενσιόμετρο γυρίσει πλήρως δεξιόστροφα, το σήμα εξόδου θα διαρκέσει περίπου 250 δευτερόλεπτα. Όταν ρυθμίζετε την ευαισθησία, είναι προτιμότερο να έχετε το χρονικό όριο όσο το δυνατόν χαμηλότερο



# Προγραμματισμός PIR αισθητήρα

8/25

Source: <https://projects.raspberrypi.org/en/projects/physical-computing/11>

```
from gpiozero import MotionSensor

pir = MotionSensor(4)

while True:
    pir.wait_for_motion()
    print("You moved")
    pir.wait_for_no_motion()
```

- ▶ Για τον προγραμματισμό του αισθητήρα, γίνεται χρήση της βιβλιοθήκης gpiozero
- ▶ `pir = MotionSensor(4)`: Μέσα στην παρένθεση γίνεται ορισμός του GPIO pin 4
- ▶ Μέσα στην επανάληψη `while True`:
  - `pir.wait_for_motion()`: Το πρόγραμμα δεν εκτελείται πέρα από αυτή την εντολή μέχρι να εντοπιστεί κίνηση
  - `pir.wait_for_no_motion()`: Αντίστοιχα το πρόγραμμα δεν εκτελείται πέρα από αυτή την εντολή μέχρι να μην εντοπίζει κίνηση

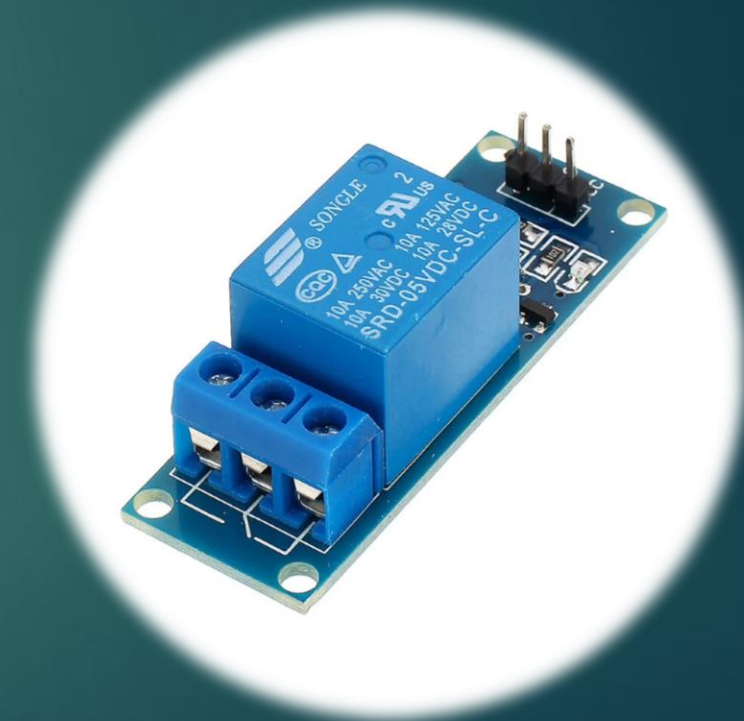


# Διακόπτης ελέγχου - ρελέ

9/25

Source: <https://tutorials-raspberrypi.com/raspberry-pi-control-relay-switch-via-gpio/>

- ▶ Μπορεί να χρειαστεί να ελέγξετε μονάδες με υψηλότερη τάση με το Raspberry Pi. Για το σκοπό αυτό, τα ρελέ μπορούν να χρησιμοποιηθούν στο Raspberry Pi. Το ρελέ χρησιμοποιείται μέσω ενός παλμού χαμηλής τάσης δεδομένου ότι το Pi ανέχεται το πολύ 5V

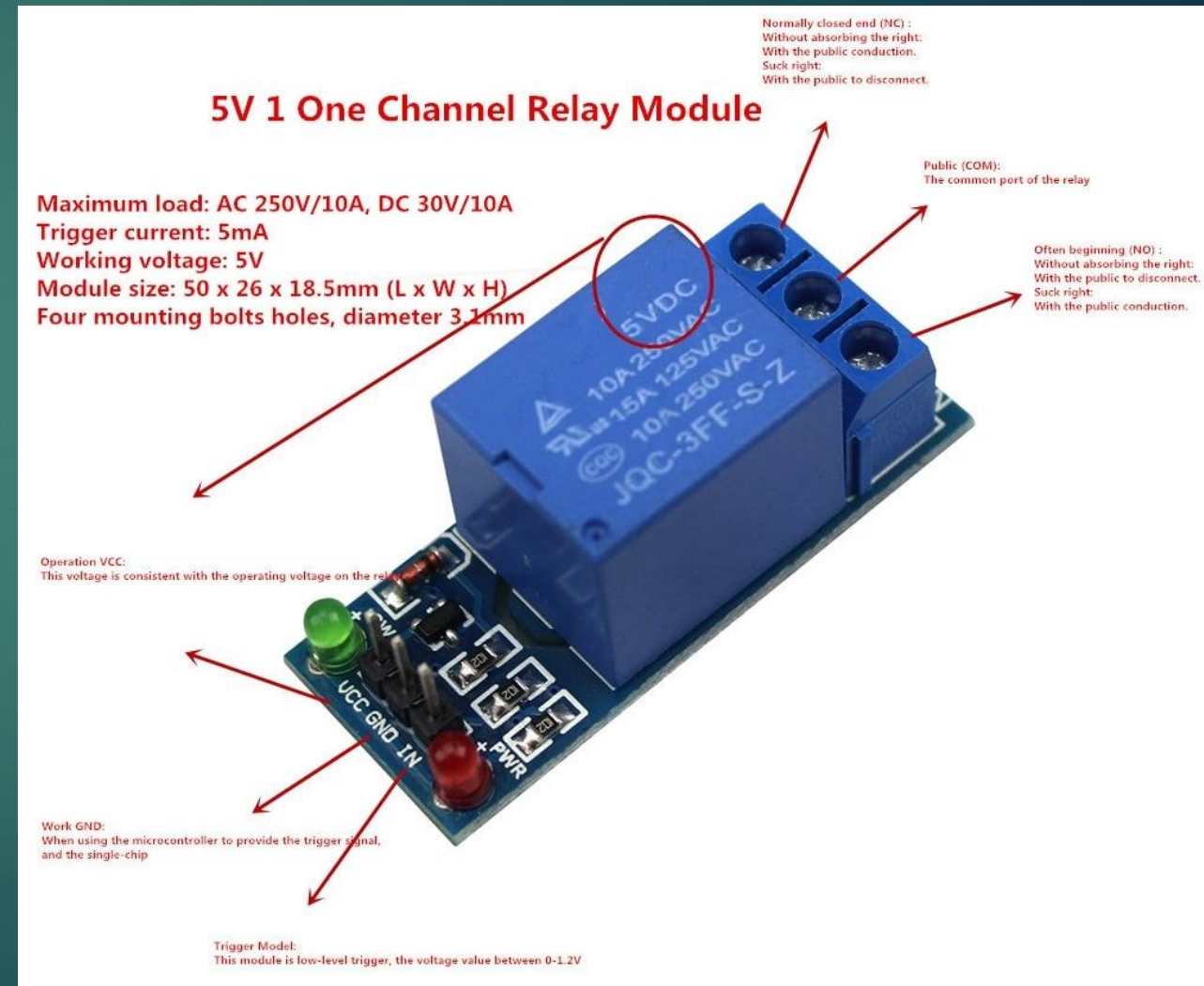


# Συνδεσμολογία ρελέ (1/3)

10/25

Source: <https://tutorials-raspberrypi.com/raspberry-pi-control-relay-switch-via-gpio/>

- ▶ Ο ακροδέκτης Vcc του ρελέ πρέπει να συνδεθεί στον ακροδέκτη 5V ή 3.3V του Raspberry Pi
- ▶ Ο ακροδέκτης Gnd του ρελέ πρέπει να συνδεθεί στον ακροδέκτη Gnd του Raspberry Pi
- ▶ Ο ακροδέκτης IN του ρελέ μπορεί να συνδεθεί σε οποιοδήποτε GPIO ακροδέκτη του Raspberry Pi

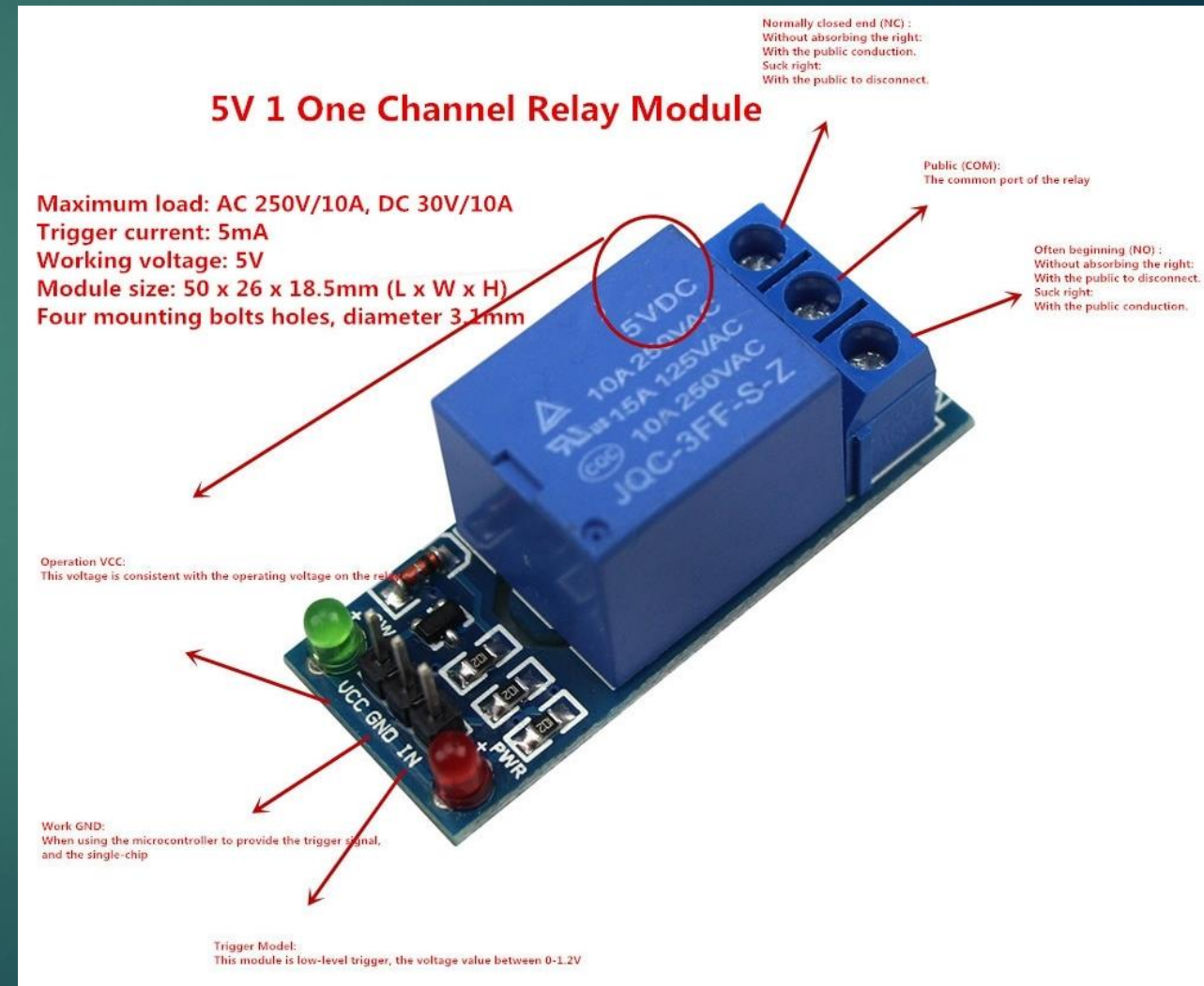


# Συνδεσμολογία ρελέ (2/3)

11/25

Source: <https://tutorials-raspberrypi.com/raspberry-pi-control-relay-switch-via-gpio/>

- ▶ Για να συνδέσετε την είσοδο του ρεύματος αρκεί να ξεβιδώσετε και να συνδέσετε το καλώδιο στον ακροδέκτη NO όπως φαίνεται στο σχήμα
- ▶ Για την έξοδο συνδέστε στον μεσαίο ακροδέκτη COM

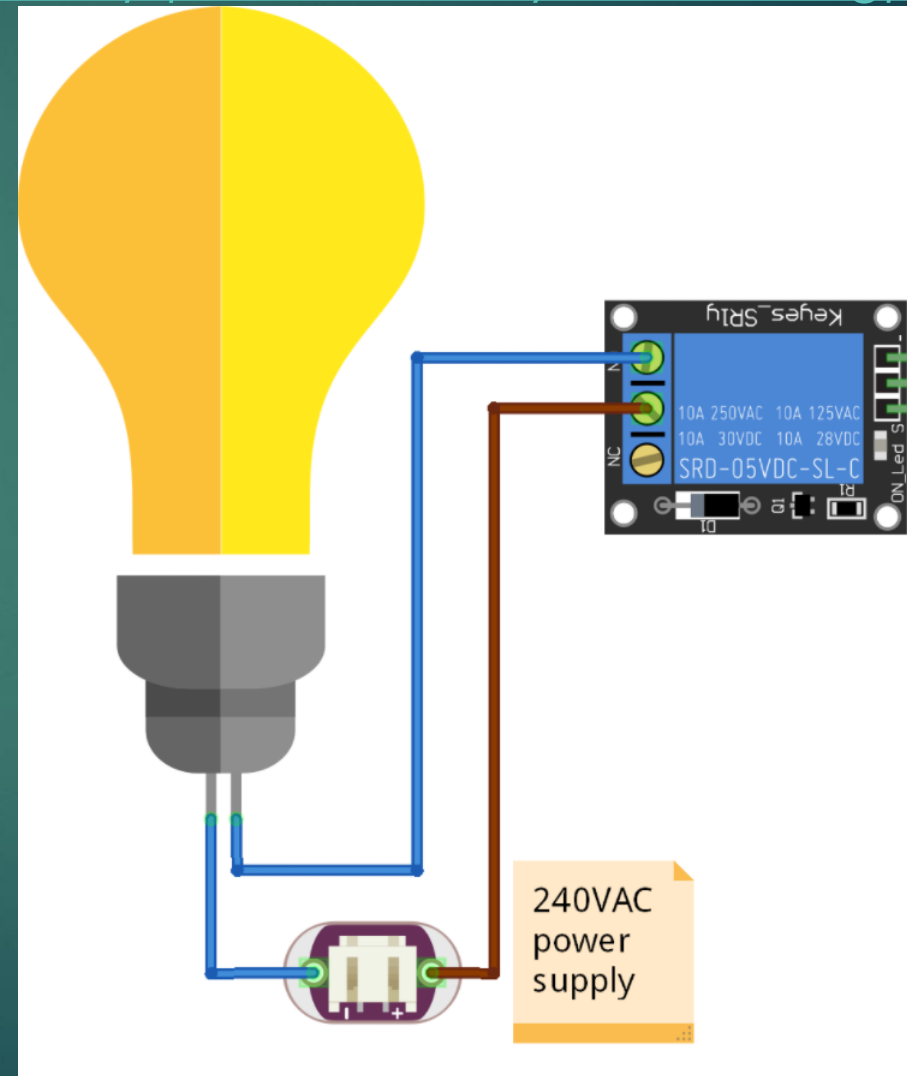


# Συνδεσμολογία ρελέ (3/3)

12/25

Source: <https://tutorials-raspberrypi.com/raspberry-pi-control-relay-switch-via-gpio/>

- ▶ Στην εικόνα φαίνεται ένα παράδειγμα για να ελέγξετε μια λάμπα



# Προγραμματισμός ρελέ

13/25

Source: <https://tutorials-raspberrypi.com/raspberry-pi-control-relay-switch-via-gpio/>

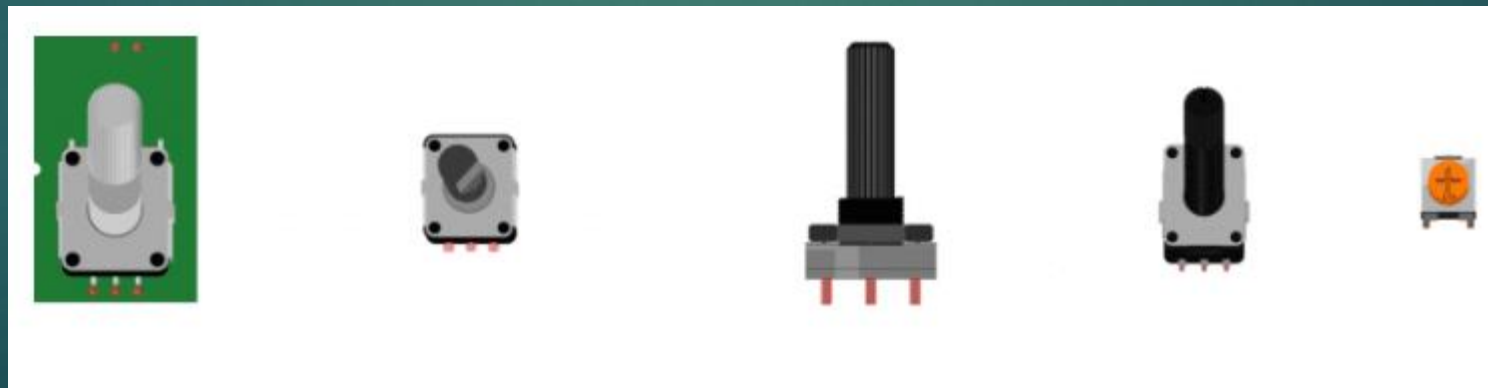
```
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(32, GPIO.OUT)
5 GPIO.output(32, GPIO.HIGH)
6 time.sleep(2)
7 GPIO.output(32, GPIO.LOW)
```

- ▶ Ο προγραμματισμός είναι πολύ απλός
- ▶ Αρκεί να χρησιμοποιήσετε την βιβλιοθήκη Rpi.GPIO
- ▶ Με την `GPIO.setmode(GPIO.BOARD)` ορίζετε τον τρόπο αρίθμησης των ακροδεκτών
- ▶ Με την `GPIO.setup(32,GPIO.OUT)` ορίζετε τον ακροδέκτη 32 ως έξοδο
- ▶ Με την `GPIO.output(32, GPIO.HIGH)` ανοίγετε το ρελέ ενώ με την `GPIO.output(32, GPIO.LOW)` κλείνετε το ρελέ

# Ποτενσιόμετρο

14/25

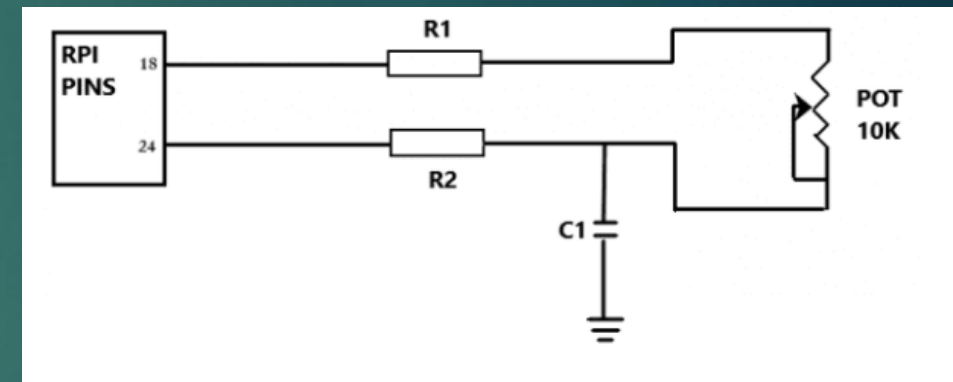
- ▶ Το ποτενσιόμετρο είναι μία μεταβλητή αντίσταση, που καθώς περιστρέφεται αυξάνεται ή μειώνεται η τιμή της αντίστασης του
- ▶ Χρησιμοποιείται για τη μεταβολή τάσης σε ένα κύκλωμα ή για τον έλεγχο ροής ρεύματος
- ▶ Έχει τρεις ακίδες (pin) εκ των οποίων το ένα χρησιμοποιείται ως είσοδος, το δεύτερο συνδέεται στη γείωση και το τρίτο ως σήμα εξόδου



# Συνδεσμολογία ποτενσιόμετρου (1/2)

15/25

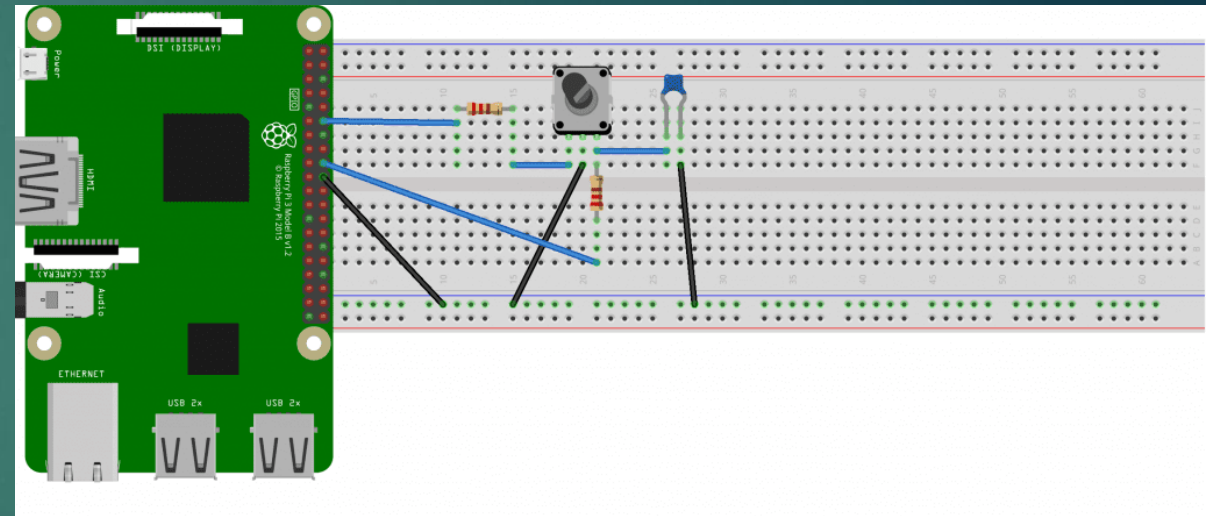
- ▶ Για τη σύνδεση ενός ποτενσιόμετρου 10KΩ με το raspberry θα χρειαστούν δύο αντιστάσεις 1KΩ και ένας πυκνωτής 220nF
- ▶ Οι δύο αντιστάσεις συνδέονται μεταξύ του ποτενσιόμετρου και των pin του raspberry
- ▶ Στη δεύτερη αντίσταση και στο ποτενσιόμετρο παρεμβάλλεται ένας πυκνωτής ο οποίος θα συνδεθεί με τη γείωση



# Συνδεσμολογία ποτενσιόμετρου (2/2)

16/25

- ▶ Στη διπλανή εικόνα παρουσιάζεται αναλυτικά η συνδεσμολογία του ποτενσιόμετρου με το raspberry
- ▶ Ακολουθείται το προηγούμενο διάγραμμα με τις αντιστάσεις και τον πυκνωτή
  - Χρησιμοποιούνται τα GPIO pin 18 και 24
  - Όπως επίσης και ένα pin γείωσης για το μεσαίο pin του ποτενσιόμετρου και τον πυκνωτή





# Κώδικας ποτενσιόμετρου

17/25

## *#Βιβλιοθήκες*

```
import RPi.GPIO as GPIO
import time
```

## *#ρύθμιση pinout*

```
GPIO.setmode(GPIO.BCM)
pin-a = 18
pin-b = 24
```

## *#ορισμός pin κατά την εκφόρτιση πυκνωτή*

```
def discharge():
    GPIO.setup(pin-a, GPIO.IN)
    GPIO.setup(pin-b, GPIO.OUT)
    GPIO.output(pin-b, False)
    time.sleep(0.004)
```

## *#ορισμός pin κατά τη φόρτιση πυκνωτή*

```
def charge_time():
    GPIO.setup(pin-b, GPIO.IN)
    GPIO.setup(pin-a, GPIO.OUT)
    count = 0
    GPIO.output(pin-a, True)
    while not GPIO.input(b_pin):
        count = count + 1
    return count
```

## *#διάβασμα τιμών*

```
def analog_read():
    discharge()
    return charge_time()
```

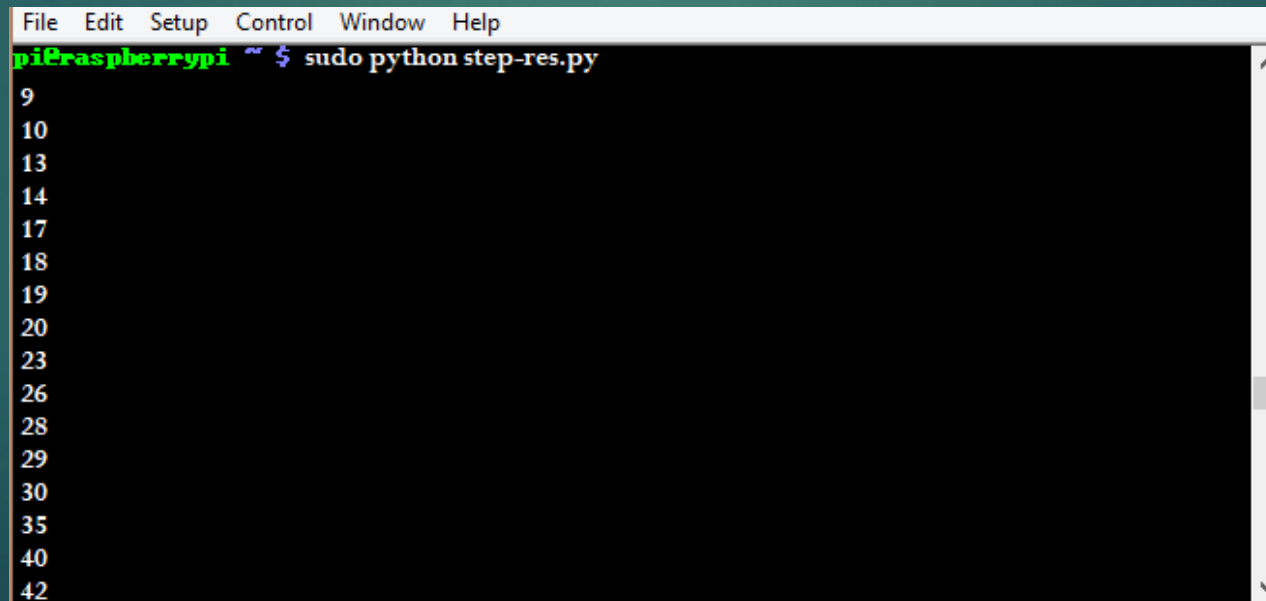
## *#εμφάνιση τιμών*

```
while True:
    print(analog_read())
    time.sleep(1)
```

# Εμφάνιση τιμών στο τερματικό

18/25

- ▶ Αφού ολοκληρώσουμε το κύκλωμα και γράψουμε τον παραπάνω κώδικα, μπορούμε να τον αποθηκεύσουμε με το όνομα `step-res.py`
- ▶ Στη συνέχεια πληκτρολογώντας στο τερματικό την εντολή: `sudo python step-res.py` θα εμφανιστούν οι τιμές που διαβάζει η πλακέτα από το ποτενσιόμετρο καθώς περιστρέφεται

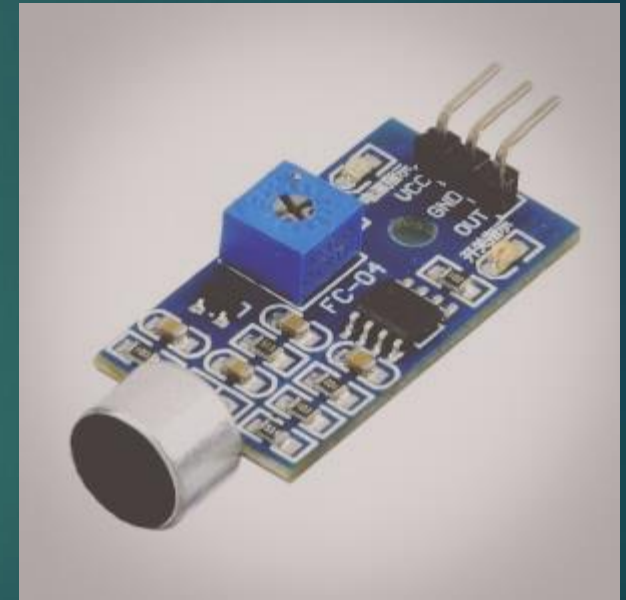


```
File Edit Setup Control Window Help
pi@raspberrypi ~ $ sudo python step-res.py
9
10
13
14
17
18
19
20
23
26
28
29
30
35
40
42
```

# Αισθητήρας ανίχνευσης ήχου

19/25

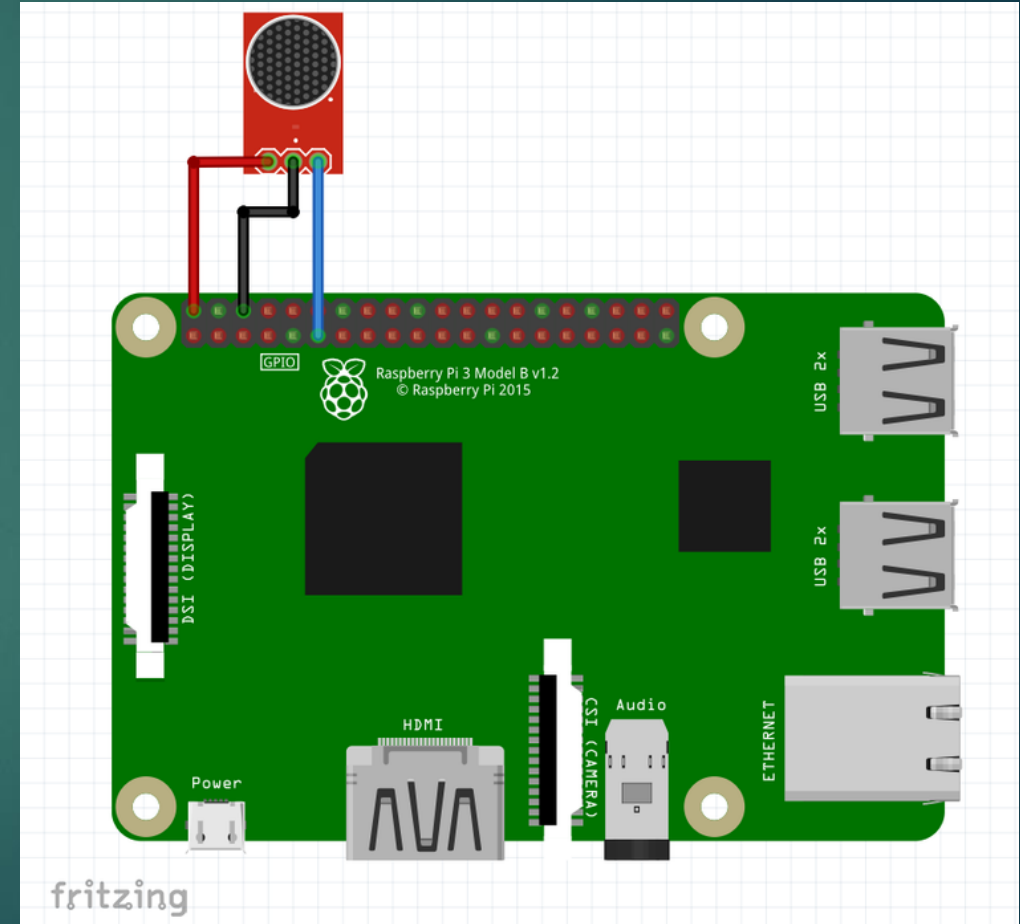
- ▶ Ο αισθητήρας ανίχνευσης ήχου ή αλλιώς sound sensor αποτελείται από ένα πυκνωτικό μικρόφωνο και συνοδεύεται από ένα ολοκληρωμένο (chip) και ποτενσιόμετρο για τη βαθμονόμηση (calibration)
- ▶ Έχει ψηφιακή έξοδο που επιστρέφει HIGH ή LOW ανάλογα με το αν εντόπισε ήχο ή όχι



# Συνδεσμολογία

20/25

- ▶ Για τη συνδεσμολογία του αισθητήρα:
  - Το Vcc θα συνδεθεί στην τάση 5v του raspberry
  - Το GND θα συνδεθεί σε ένα από τα pin γείωσης
  - Το D0 είναι η ψηφιακή έξοδος του αισθητήρα και θα συνδεθεί σε κάποιο από τα GPIO pin της πλακέτας (GPIO 17)



# Κώδικας

21 / 25

## *#Βιβλιοθήκες*

```
import RPi.GPIO as GPIO
import time
```

## *# ρύθμιση pinout*

```
channel = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel, GPIO.IN)
```

## *#εμφάνιση κατάλληλου μηνύματος*

```
def callback(channel):
    if GPIO.input(channel):
        print "Sound Detected!"
    else:
        print "Sound Not Detected!"
```

## *#αναγνώριση αλλαγής HIGH/LOW*

```
GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)
```

## *#κλήση συνάρτηση εμφάνισης μηνύματος*

```
GPIO.add_event_callback(channel, callback)
```

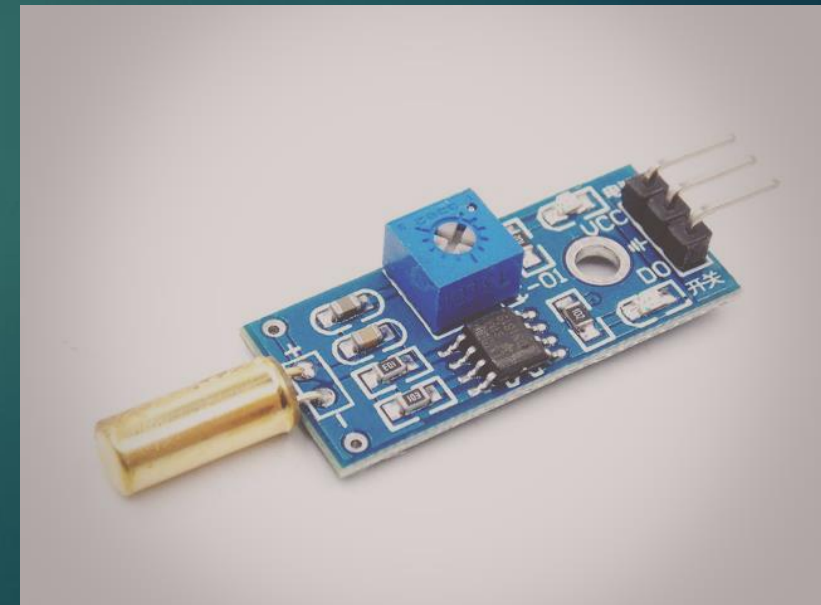
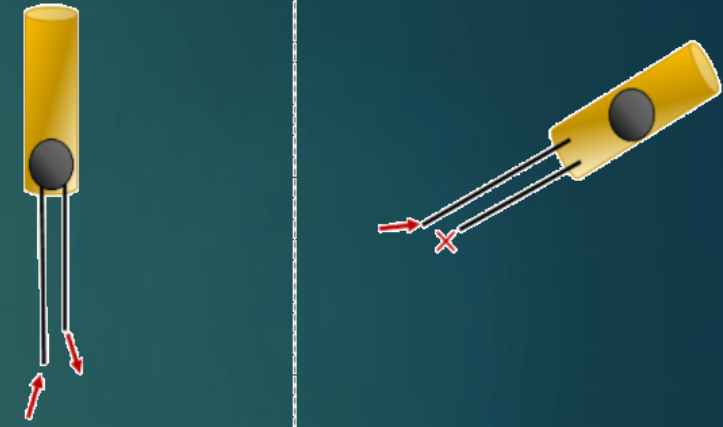
## *# επανάληψη*

```
while True:
    time.sleep(1)
```

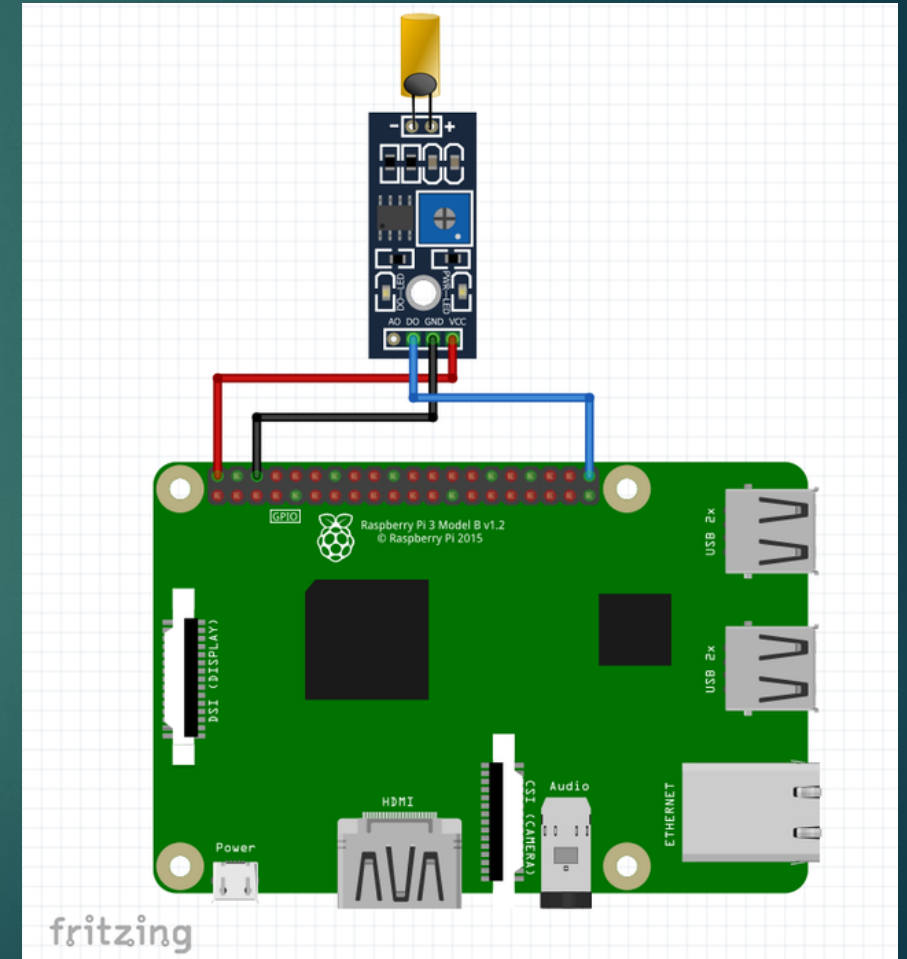
# Αισθητήρας κλίσης

22/25

- ▶ Ο αισθητήρας κλίσης ή tilt sensor αποτελείται από μία μεταλλική μπίλια με τη βοήθεια της οποίας κλείνει κύκλωμα και περνάει ρεύμα όταν βρίσκεται σε κάθετη θέση
- ▶ Όταν βρίσκεται σε οριζόντια θέση μετακινείται η μπίλια και δεν περνάει ρεύμα από τις ακίδες
- ▶ Ο αισθητήρας συνοδεύεται από μια πλακέτα που περιέχει ολοκληρωμένο (chip) και ποτενσιόμετρο για τη βαθμονόμηση του αισθητήρα



- ▶ Για τη συνδεσμολογία του αισθητήρα:
  - Το Vcc θα συνδεθεί στην τάση 5v του raspberry
  - Το GND θα συνδεθεί σε ένα από τα pin γείωσης
  - Το D0 είναι η ψηφιακή έξοδος του αισθητήρα και θα συνδεθεί σε κάποιο από τα GPIO pin της πλακέτας (GPIO 21)



# Κώδικας

24/25

```
#εισαγωγή βιβλιοθήκης  
import RPi.GPIO as GPIO
```

```
#δήλωση pin
```

```
channel = 21
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(channel, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

```
#μήνυμα ειδοποίησης
```

```
def alert(ev=None):
```

```
    print("Tilt Detected")
```

```
#διάβασμα από αισθητήρα σε επανάληψη
```

```
def loop():
```

```
    GPIO.add_event_detect(channel, GPIO.FALLING, callback=alert, bouncetime=100)
```

```
    while True:
```

```
        pass
```

```
if __name__ == '__main__':  
    try:  
        loop()  
    except KeyboardInterrupt:  
        GPIO.cleanup()
```



# Βιβλιογραφία

- [1] <https://projects.raspberrypi.org/en/projects/physical-computing/11>
- [2] <https://tutorials-raspberrypi.com/raspberry-pi-control-relay-switch-via-gpio/>
- [3] <https://www.circuitbasics.com/using-potentiometers-with-raspberry-pi/>
- [4] <https://www.instructables.com/Tilt-Sensor-Raspberry-Pi/>
- [5] <https://robu.in/tilt-sensor-with-raspberry-pi/>
- [6] <https://www.instructables.com/Sound-Sensor-Raspberry-Pi/>