Εισαγωγή σε Αισθητήρες - Συνδεσμολογία

ΜΑΘΗΜΑΤΑ RASPBERRY

ΜΑΝΟΣ ΝΙΚΟΛΑΟΣ

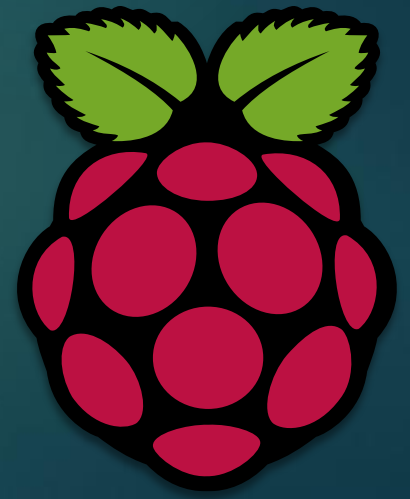
ΔΟΥΜΑ ΑΝΑΣΤΑΣΙΑ

ΚΑΒΑΛΛΙΕΡΑΤΟΥ ΕΡΓΙΝΑ

Περιεχόμενα

Εισαγωγή στους Αισθητήρες

- Η χρήση των αισθητήρων είναι απαραίτητη στις περισσότερες εφαρμογές ρομποτικών συστημάτων ή συστημάτων αυτοματισμού
- Κάθε αισθητήρας μετράει κάποιες τιμές (θερμοκρασία, απόσταση, φωτεινότητα, κ.α.) από το εξωτερικό περιβάλλον και τις στέλνει στην πλακέτα όπου είναι συνδεδεμένος
- Στη συνέχεια μέσω της πλακέτας γίνεται η επεξεργασία των τιμών και ανάλογα με αυτές το πρόγραμμα εκτελεί τις κατάλληλες λειτουργίες



Κατηγορίες αισθητήρων (1/2)

- Οι αισθητήρες μπορούν να ταξινομηθούν σε **ιδιοδεκτικούς**, δηλαδή σε αυτούς που μετράνε τιμές εσωτερικά του συστήματος και **εξωδεκτικούς**, δηλαδή αυτούς που μετράνε τιμές από το εξωτερικό περιβάλλον

- Μπορούν να διακριθούν σε **αναλογικούς** όπου επιστρέφουν ένα εύρος τιμών ή σε **ψηφιακούς** όπου επιστρέφουν 0 ή 1

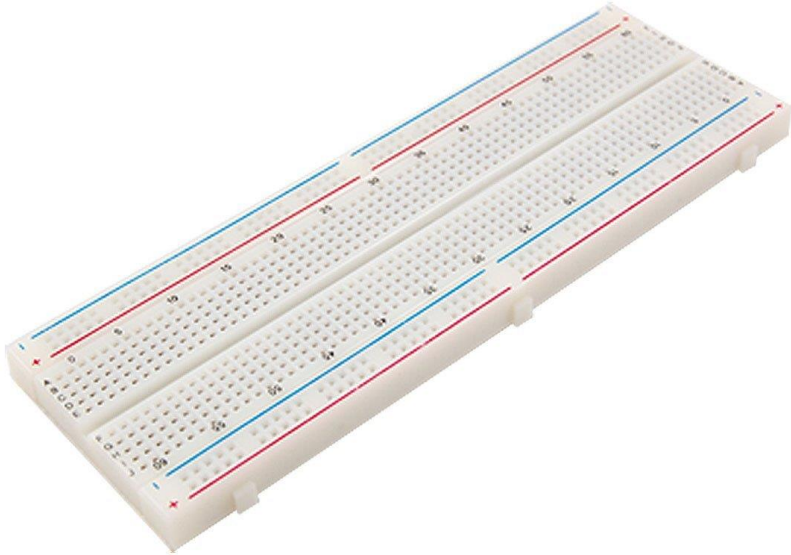
Κατηγορίες αισθητήρων (2/2)

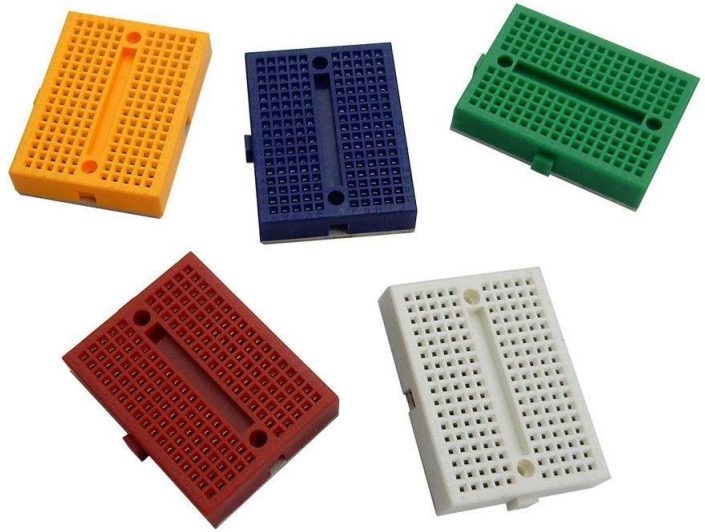
- Τέλος, οι αισθητήρες μπορούν να διακριθούν σε **παθητικούς**, δηλαδή μετράνε την ενέργεια που έρχεται από το περιβάλλον και **ενεργητικούς**, δηλαδή εκπέμπουν την κατάλληλη ενέργεια και μετρούν την αντίδραση

Συνδεσμολογία (1/4)

- Το breadboard ή πλακέτα δοκιμών χρησιμοποιείται για τη σύνδεση των αισθητήρων, των κινητήρων και οποιωνδήποτε άλλων ηλεκτρονικών στοιχείων (αντιστάσεις, πυκνωτές, κ.α.)

- Υπάρχουν διάφορες διατάξεις και μεγέθη breadboard ανάλογα με το κύκλωμα που θα χρειαστεί να υλοποιηθεί

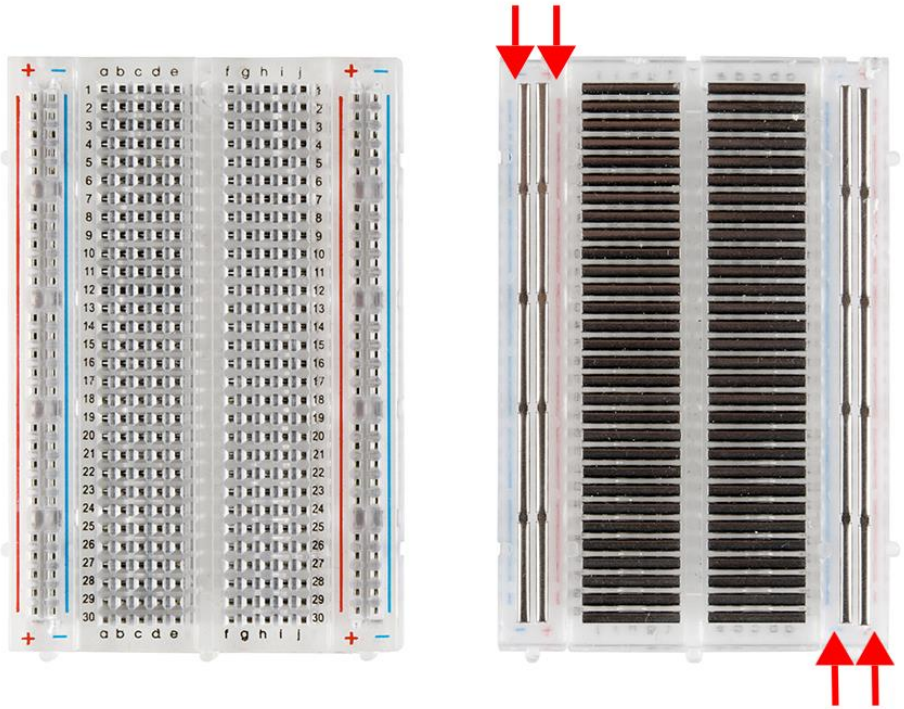




Συνδεσμολογία (2/4)

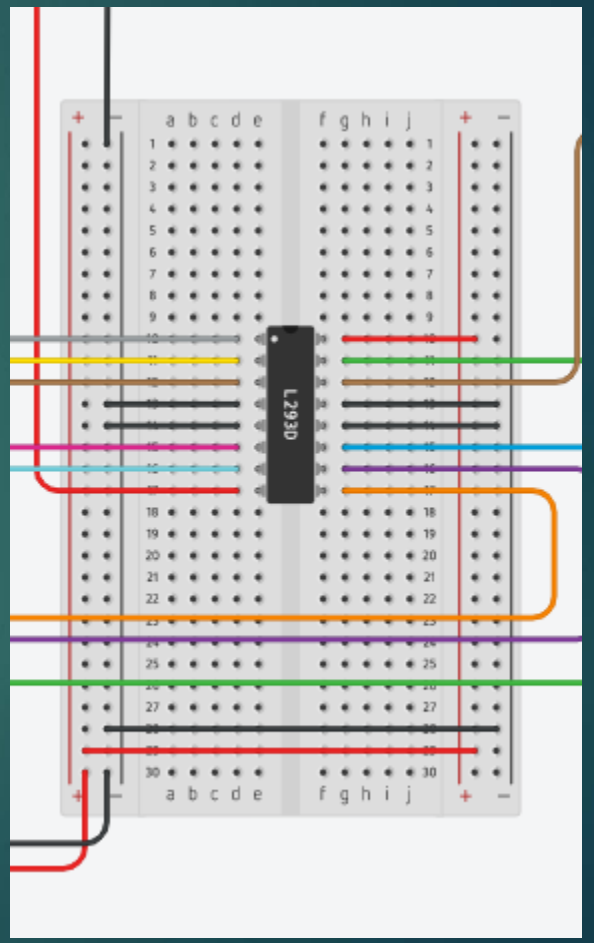
- Κάθε breadboard διαθέτει οριζόντιες και τις περισσότερες φορές και κάθετες βραχυκυκλωμένες γραμμές

- Οι οριζόντιες γραμμές βοηθάνε στις συνδέσεις των ηλεκτρονικών στοιχείων, ενώ οι κάθετες στις τάσεις θετικές (+) και αρνητικές (-) ή γειώσεις



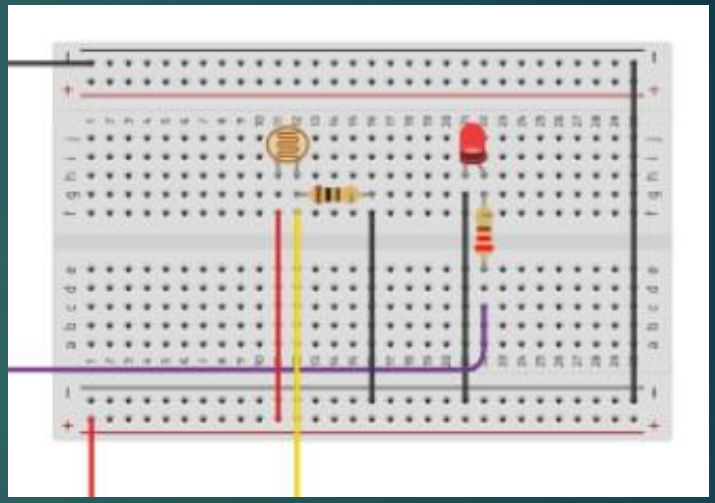
Συνδεσμολογία (3/4)

- Κάθε breadboard διαθέτει μια κάθετη γραμμή στο κέντρο, η οποία βοηθάει στην τοποθέτηση ολοκληρωμένων (chip, π.χ. L293D)
- Δεξιά και αριστερά της γραμμής αυτής σχηματίζονται δύο περιοχές (a,b,c,d,e και f,g,h,i,j) οι οποίες δεν βραχυκυκλώνονται μεταξύ τους



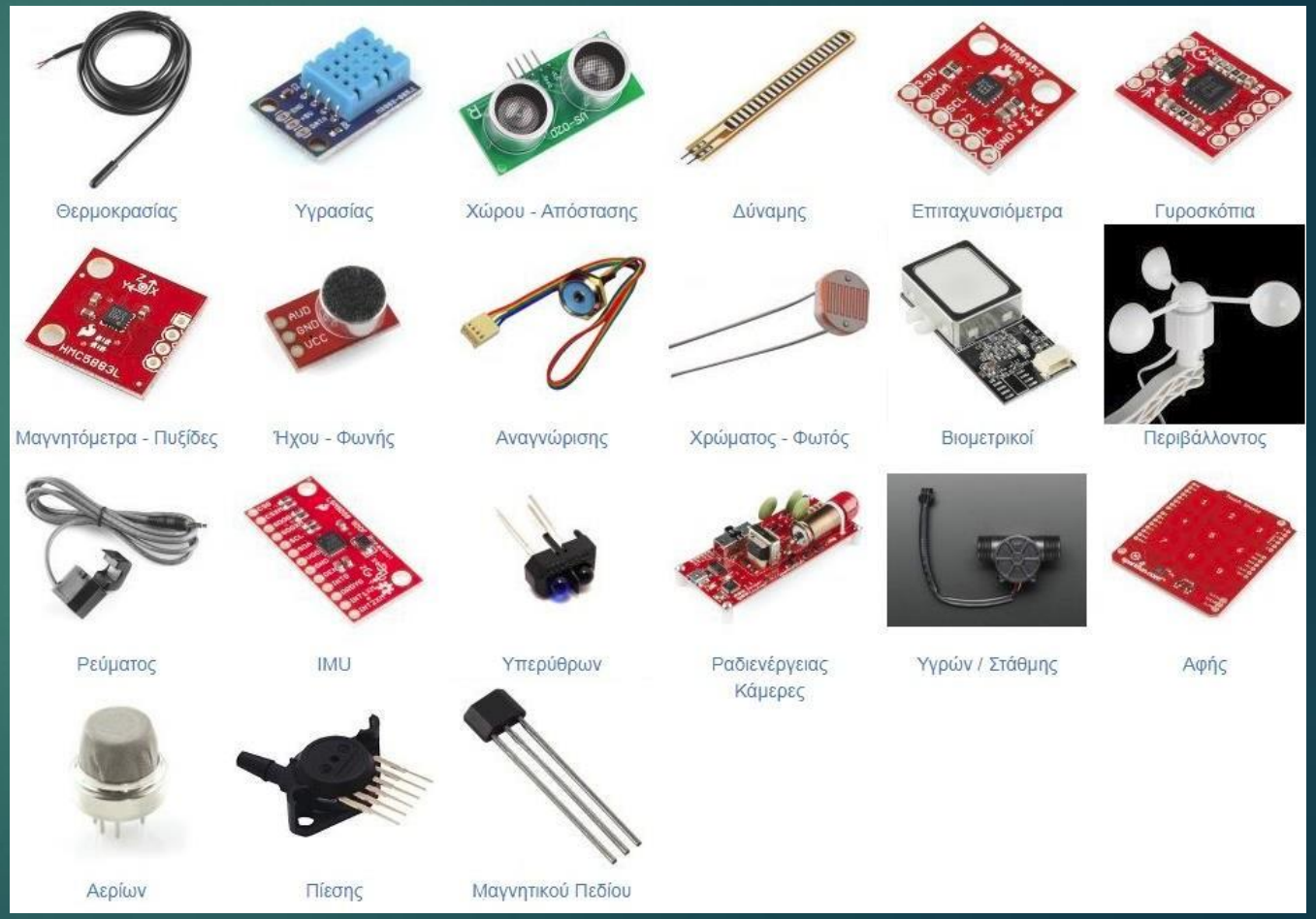
Συνδεσμολογία (4/4)

- **Χρωματισμός καλωδίων:**
  - Με **κόκκινο** χρώμα συμβολίζονται οι καλωδιώσεις των τάσεων (3.3V, 5V)
  - Με **μαύρο** χρώμα οι γειώσεις (GND ή -)
  - Όλα τα υπόλοιπα καλώδια μπορούν να συμβολιστούν με όποιο χρωματισμό επιθυμεί κάποιος, εκτός των δύο παραπάνω για να είναι διακριτά και να μη γίνει κάποιο λάθος στις συνδέσεις



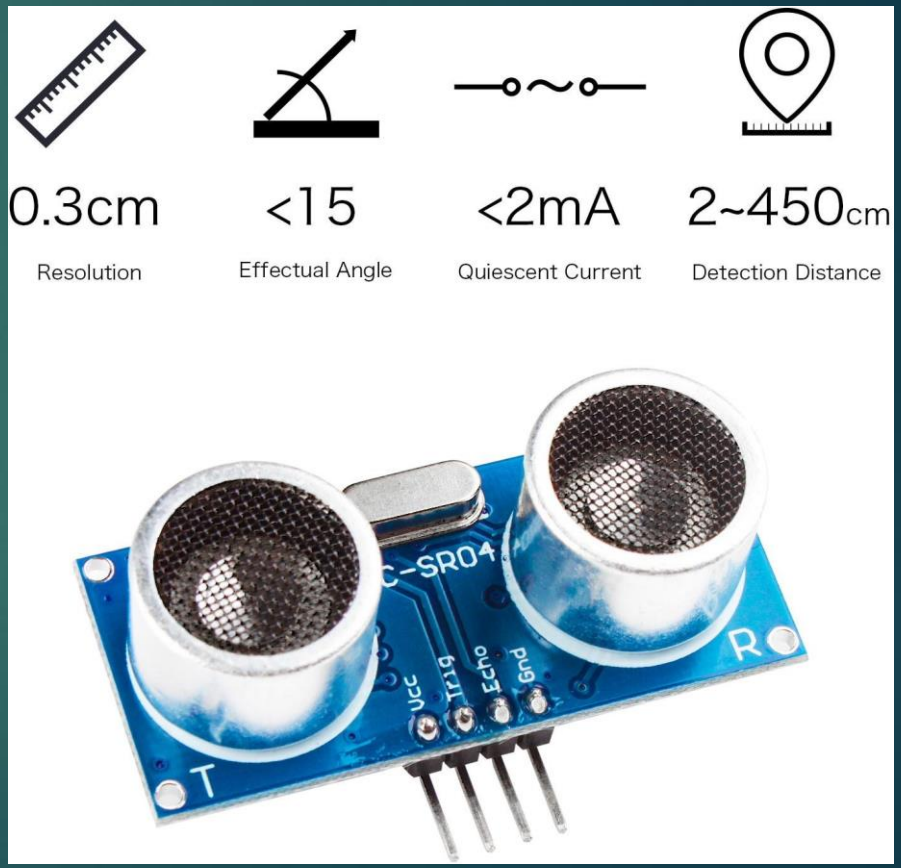
Τύποι Αισθητήρων

- Υπάρχουν διάφοροι τύποι αισθητήρων:
  - αισθητήρες προσέγγισης
  - αισθητήρες θέσης
  - αισθητήρες ρευστών
  - αισθητήρες θερμοκρασίας
  - αισθητήρες φωτεινότητας
  - αισθητήρες πίεσης
  - αισθητήρες επιτάχυνσης



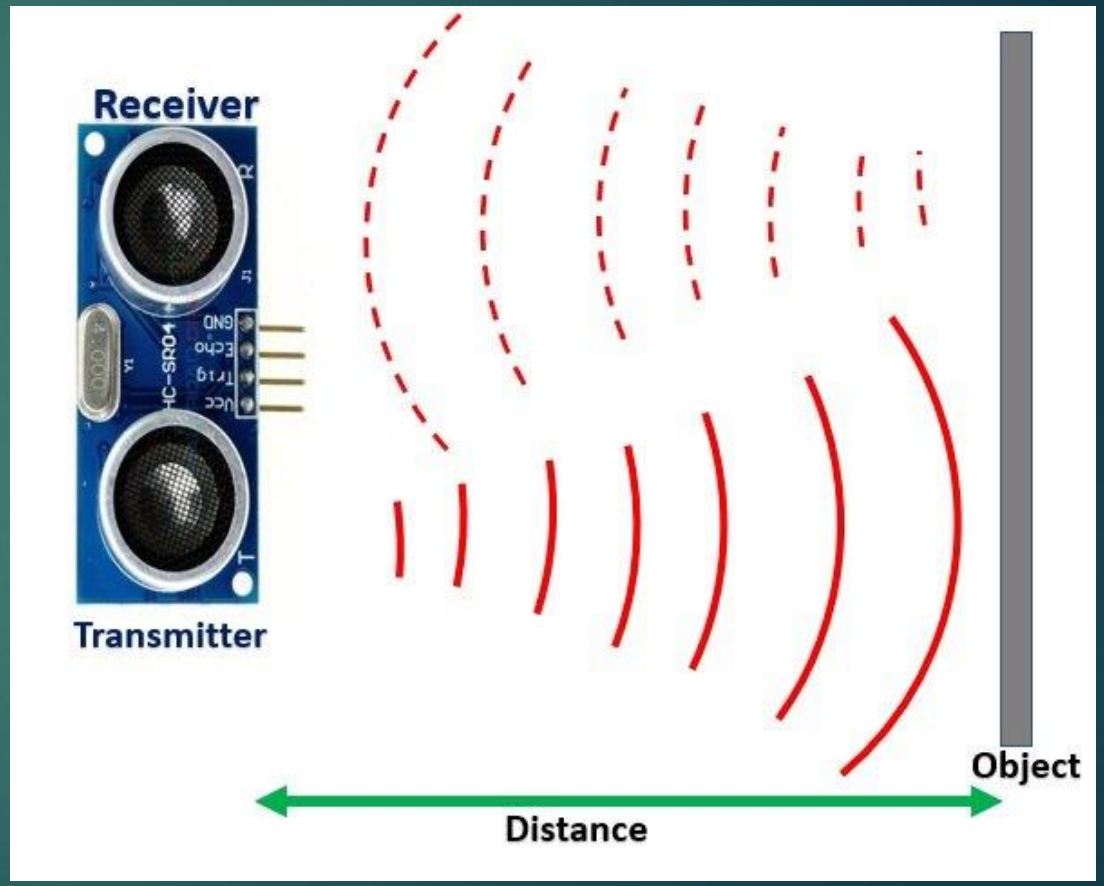
Αισθητήρας Ultrasonic (1/4)

- Ο αισθητήρας απόστασης υπερήχων (ultrasonic) έχει **ακρίβεια** 0.3 cm
- Έχει **απόδοση** σε γωνία μικρότερη των 15 μοιρών
- Το **ρεύμα** που καταναλώνει είναι λιγότερο από 2mA
- Μπορεί να μετρήσει **αποστάσεις** μεταξύ 2cm – 450cm



Αισθητήρας Ultrasonic (2/4)

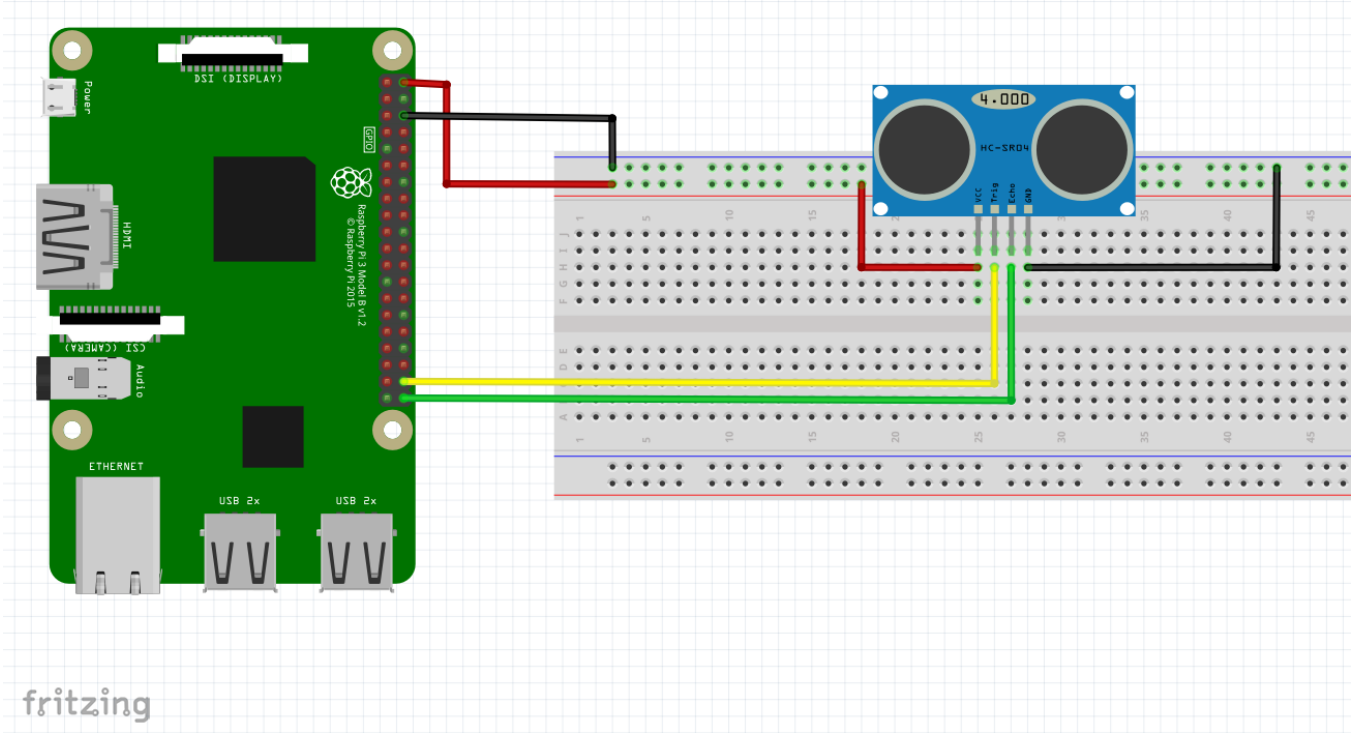
- Αποτελείται από έναν πομπό και έναν δέκτη υπερήχων, όπως φαίνεται στην εικόνα
- Καθώς στέλνει ο πομπός ένα σήμα υπερήχου, αντανακλάται στην επιφάνεια που υπάρχει μπροστά από τον αισθητήρα και με βάση την καθυστέρηση επιστροφής στο δέκτη υπολογίζεται η απόσταση



Αισθητήρας Ultrasonic (3/4)

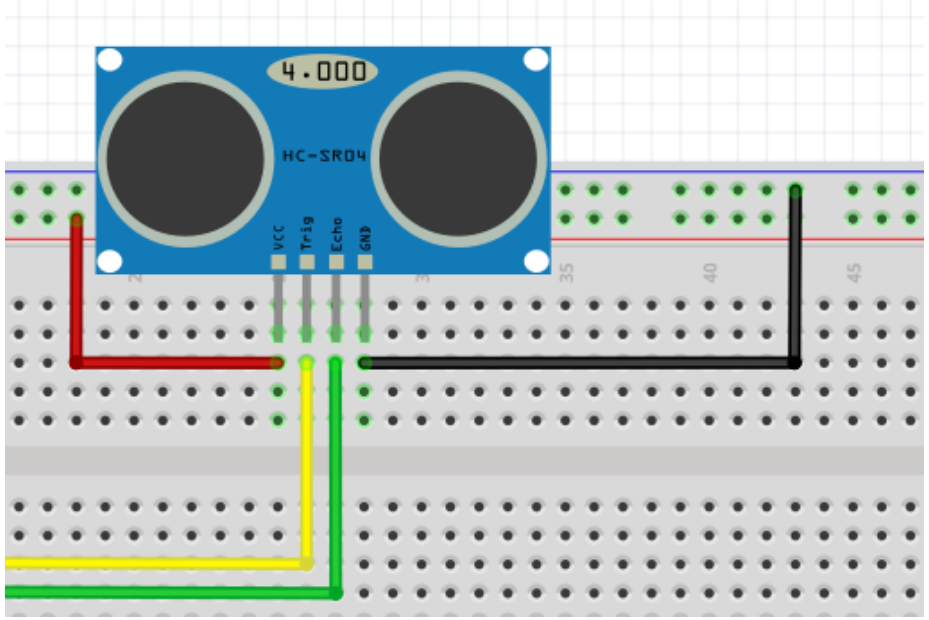
Διαθέτει 4 pin εκ των οποίων:

1. Το **Vcc** συνδέεται με την τάση στα 5v του Raspberry
2. Το **trig** αντιστοιχεί στο trigger και είναι το σήμα που θα στείλει ο αισθητήρας για τον υπέρηχο
3. Το **echo** είναι το σήμα που θα λάβει ο αισθητήρας από την επιστροφή του υπέρηχου
4. Το **GND** συνδέεται σε κάποιο από τα pin γείωσης



Αισθητήρας Ultrasonic (4/4)

- Τόσο το echo όσο και το trigger θα συνδεθούν σε δύο από τα σήματα GPIO της πλακέτας
- Ιδιαίτερη προσοχή πρέπει να δοθεί στην καλωδίωση αλλά και στον κώδικα ώστε να γίνει σωστή αντιστοιχία των δύο παραπάνω σημάτων, διαφορετικά δεν θα δουλέψει ο αισθητήρας



Κώδικας (1/2)


```
#Βιβλιοθήκες
import RPi.GPIO as GPIO
import time

#GPIO λειτουργία (BOARD / BCM)
GPIO.setmode(GPIO.BCM)

#ορισμός GPIO Pins
GPIO_TRIGGER = 20
GPIO_ECHO = 21

#ορισμός GPIO εισόδου/εξόδου
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
```

```
def distance():
    # ορισμός Trigger σε HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # ορισμός Trigger μετά από 0.01ms σε LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()

    # αποθήκευση χρόνου έναρξης
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()

    # αποθήκευση χρόνου άφιξης
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()
```

Κώδικας (2/2)

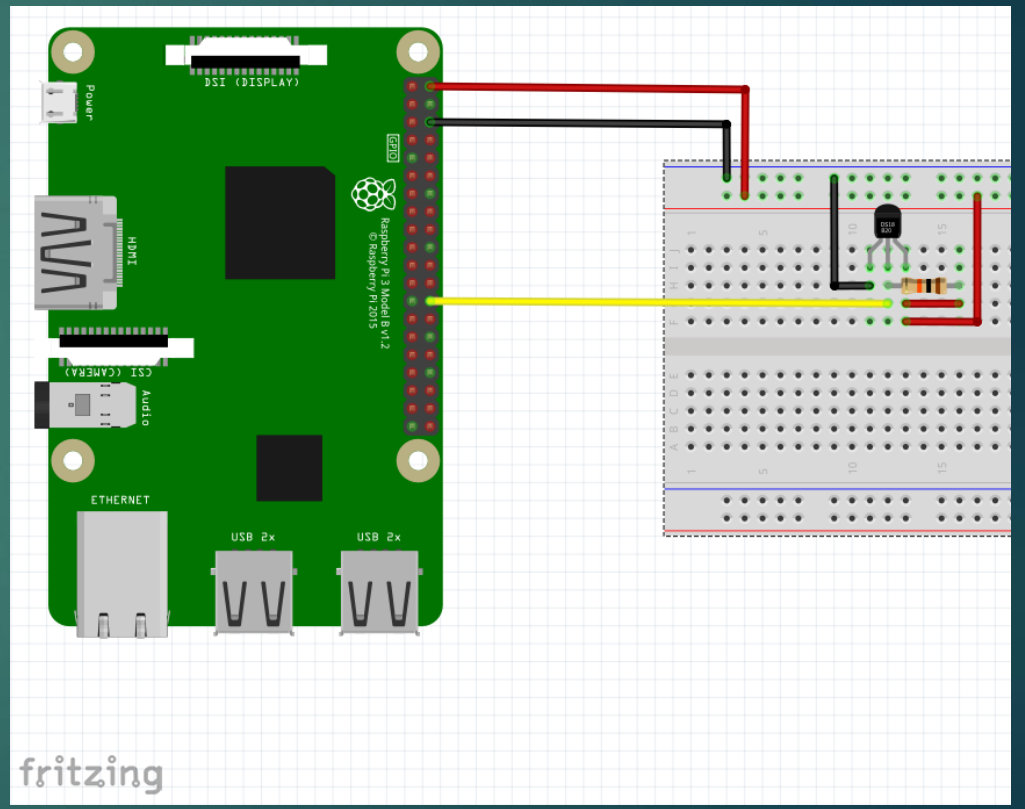

```
# διαφορά χρόνου μεταξύ έναρξης και άφιξης
    TimeElapsed = StopTime - StartTime
    # πολ/σμός με την ταχύτητα του ήχου (34300 cm/s)
    # διαίρεση με το 2, για την αποστολή και την παραλαβή
    distance = (TimeElapsed * 34300) / 2
    return distance

#εμφάνιση απόστασης
if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print ("Measured Distance = %.1f cm" % dist)
            time.sleep(1)

        # εκκαθάριση GPIO
            GPIO.cleanup()
```

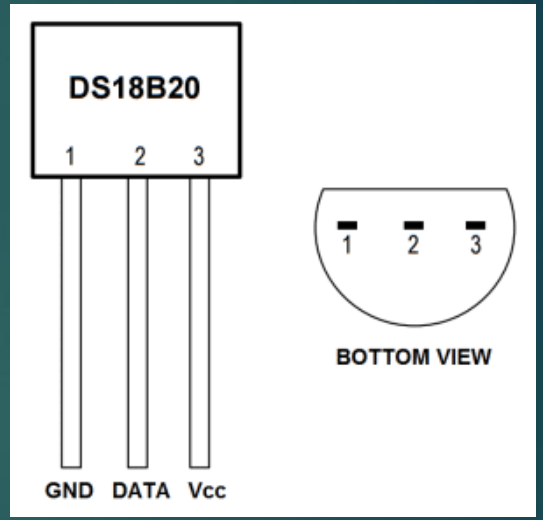
Αισθητήρας Θερμοκρασίας DS18B20 (1/2)

- Η συνδεσμολογία του αισθητήρα θερμοκρασίας DS1820 παρουσιάζεται στη διπλανή εικόνα
- Θα χρειαστεί μία αντίσταση 10ΚΩ μεταξύ της τάσης και του σήματος
- Το τρανζίστορ έχει ψηφιακή έξοδο, οπότε δε θα χρειαστεί κάποια μετατροπή για την πλακέτα



Αισθητήρας Θερμοκρασίας DS18B20 (2/2)

- Πιο αναλυτικά:
  - Το **pin 1** συνδέεται στην γείωση της πλακέτας
  - Το **pin 2** συνδέεται σε μία GPIO θύρα του Raspberry ώστε να διαβάζει το σήμα από τον αισθητήρα και το ένα άκρο μιας αντίστασης 10KΩ
  - Το **pin 3** συνδέεται στην τάση 5V και το δεύτερο άκρο της αντίστασης



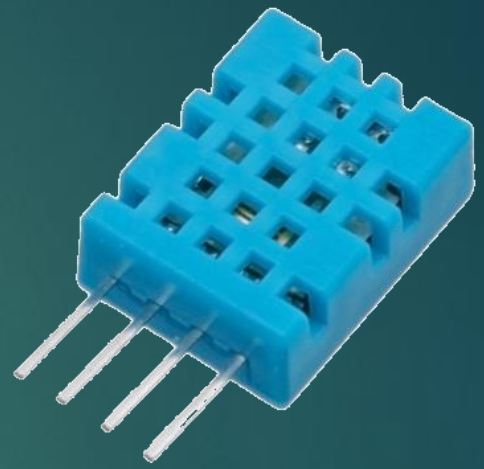
Κώδικας


```
#εισαγωγή βιβλιοθηκών
import time
from w1thermsensor import W1ThermSensor

#δημιουργία αντικειμένου sensor
sensor = W1ThermSensor()

#διάβασμα και εκτύπωση θερμοκρασιών
while True:
    temperature = sensor.get_temperature()
    print("The temperature is %s celsius" % temperature)
    time.sleep(1)
```

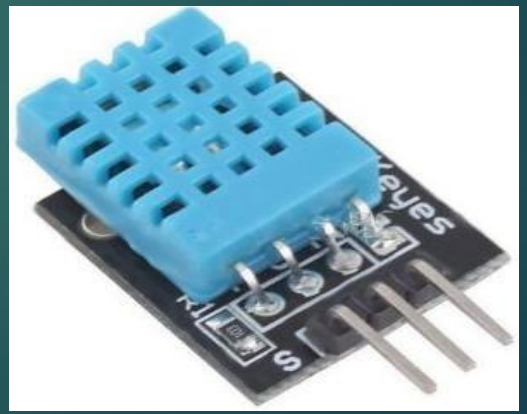
Αισθητήρας Υγρασίας (1/2)



- Ο αισθητήρας DHT11 μετράει την υγρασία του περιβάλλοντος στο οποίο βρίσκεται, αλλά και τη θερμοκρασία

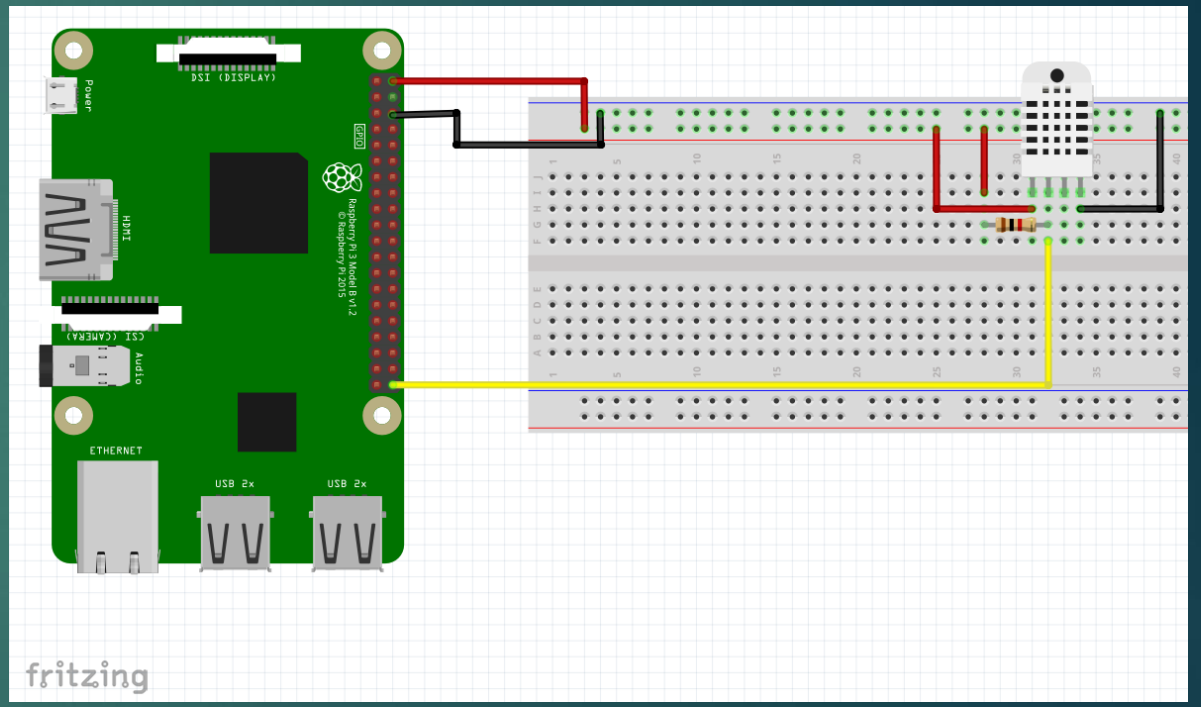
- Είναι ψηφιακός αισθητήρας και αποτελείται από 4 pin εκ των οποίων το ένα δεν χρειάζεται να συνδεθεί στην πλακέτα

- Στην περίπτωση που συνοδεύεται από το module όπως φαίνεται στην κάτω εικόνα δεν χρειάζεται επιπλέον αντίσταση στο κύκλωμα



Αισθητήρας Υγρασίας (2/2)

- Για τη συνδεσμολογία του αισθητήρα υγρασίας:
  - Το **pin 1** θα συνδεθεί στην τάση 5V του Raspberry
  - Το **pin 2** θα συνδεθεί σε κάποια από τις GPIO θύρες του Raspberry και επιπλέον σε μία αντίσταση 1ΚΩ, ενώ το άλλο άκρο της θα συνδεθεί στην τάση
  - Το **pin 3** δεν θα χρειαστεί να συνδεθεί
  - Το **pin 4** θα συνδεθεί στη γείωση



Κώδικας (1/2)


```
#εισαγωγή βιβλιοθηκών
import time
import board
import adafruit_dht
# αρχικοποίηση του αισθητήρα, και αντιστοίχιση με τη θύρα σύνδεσης
dhtDevice = adafruit_dht.DHT11(board.D21, use_pulseio=False)
while True:
    try:
        # εκτύπωση τιμών θερμοκρασίας και υγρασίας
        temperature_c = dhtDevice.temperature
        temperature_f = temperature_c * (9 / 5) + 32
        humidity = dhtDevice.humidity
        print("Temp: {:.1f} F / {:.1f} C    Humidity: {}% ".format(
                temperature_f, temperature_c, humidity
            )
        )
```

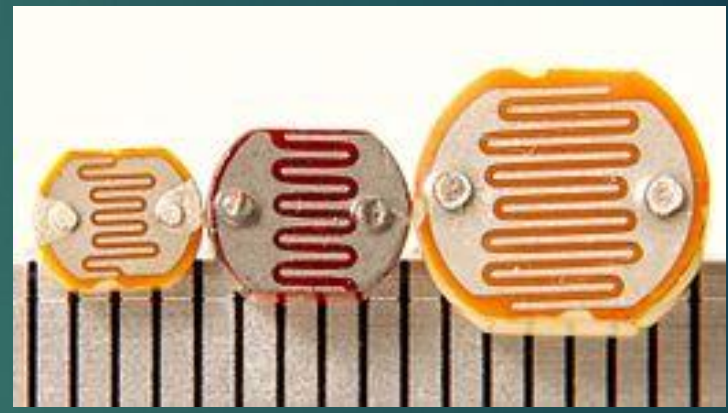
Κώδικας (2/2)


```
except RuntimeError as error:
        # σε περίπτωση λάθους
        print(error.args[0])
        time.sleep(2.0)
        continue
    except Exception as error:
        dhtDevice.exit()
        raise error
         time.sleep(2.0)
```

Αισθητήρας Φωτός-Φωτοαντίσταση

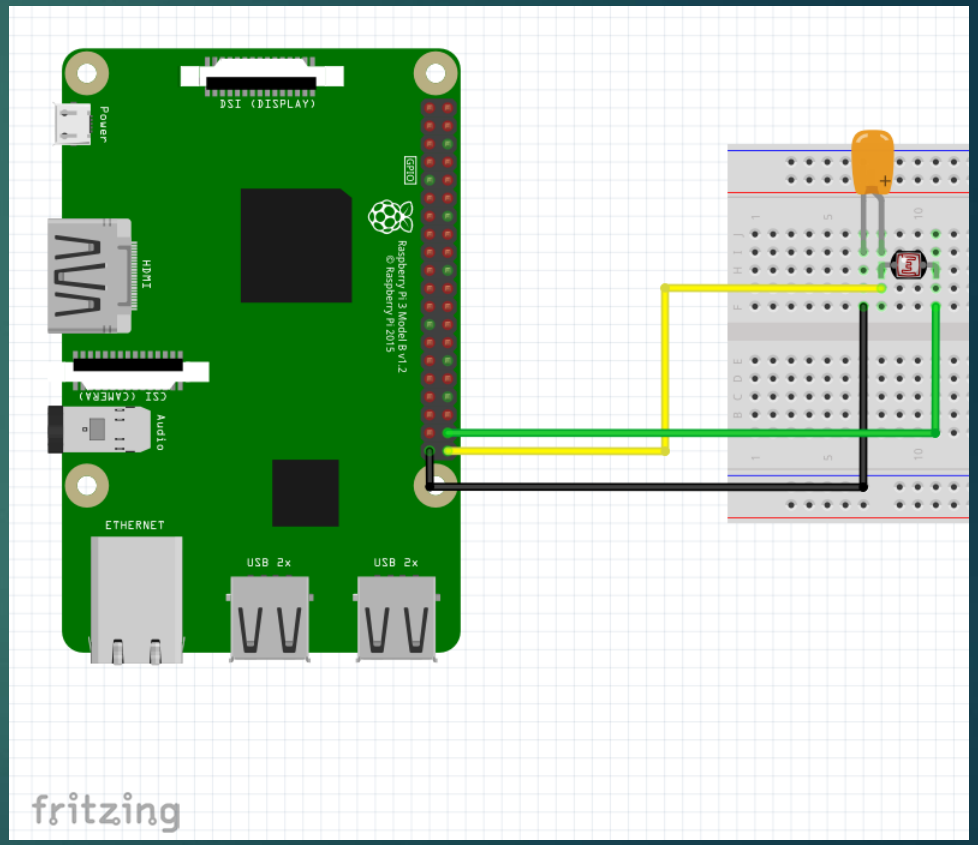
- Ένας αισθητήρας φωτός ή φωτοαντίσταση επιστρέφει αναλογικό σήμα στην πλακέτα, όμως το Raspberry έχει μόνο ψηφιακές εισόδους/εξόδους

- Επομένως, μπορεί να χρησιμοποιηθεί ένας πυκνωτής 1μF στο κύκλωμα ώστε η πλακέτα να αναγνωρίζει 2 καταστάσεις, HIGH ή LOW



Αισθητήρας Φωτός-Φωτοαντίσταση

- Για τη συνδεσμολογία:
  - Το ένα pin του πυκνωτή συνδέεται στο GND του Raspberry
  - Το δεύτερο pin του πυκνωτή μαζί με το ένα pin του photoresistor συνδέεται σε μία GPIO θύρα της πλακέτας
  - Το δεύτερο pin του photoresistor συνδέεται σε μια δεύτερη θύρα της πλακέτας



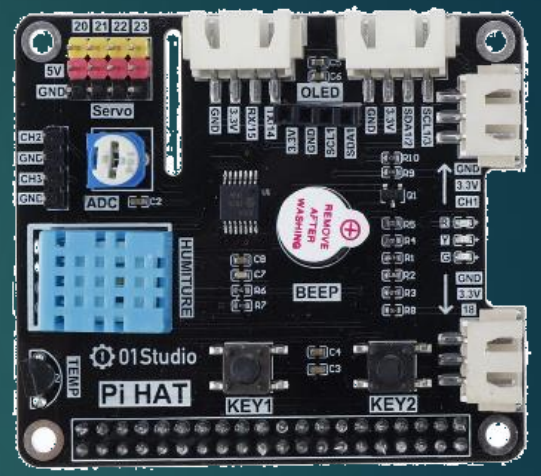
Κώδικας


```
#εισαγωγή βιβλιοθηκών και δήλωση pin
import RPi.GPIO as GPIO
import time
readpin=20
tpin=21
#ορισμός των pin ως GPIO
GPIO.setmode(GPIO.BCM)
#ορισμός του pin για έξοδο
GPIO.setup(tpin,GPIO.OUT)
GPIO.setwarnings(False)
#ορισμός του pin για είσοδο και διάβασμα τιμής
GPIO.setup(readpin,GPIO.IN)
GPIO.setwarnings(True)
print (" Read: " + str(GPIO.input(readpin)))
time.sleep(1)
```

Επεκτάσεις

- **Hat Raspberry**

  - Κάποιοι από τους παραπάνω αισθητήρες υπάρχουν και σε μορφή πλακέτας επέκτασης(Hat) που συνδέεται στο Raspberry
  - Σε αυτή την περίπτωση δεν χρειάζεται η συνδεσμολογία και η καλωδίωση με breadboard, διότι συνδέεται άμεσα το Raspberry με το Hat και κατ' επέκταση και οι αισθητήρες που υπάρχουν στο Hat
  - Επομένως, θα χρειαστεί μόνο το προγραμματιστικό κομμάτι



Προσοχή σε:

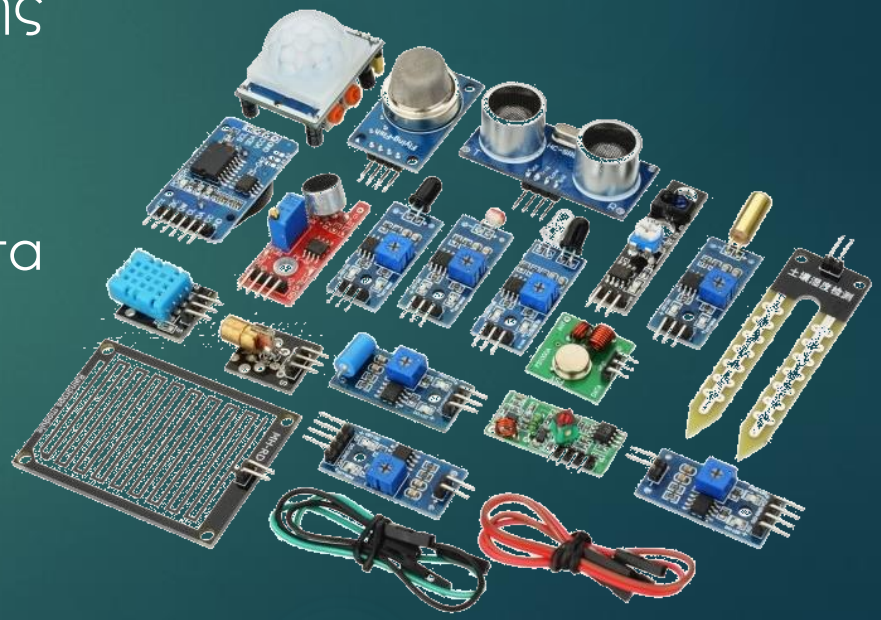
- **Τροφοδοσία**
  - Οι περισσότεροι αισθητήρες λειτουργούν στα 3,3V ή 5V, για το λόγο αυτό χρειάζεται η σωστή συνδεσμολογία στην τάση. Σε περίπτωση που η τάση λειτουργείας κάποιου αισθητήρα είναι διαφορετική των 2 παραπάνω, τότε μπορούν να χρησιμοποιηθούν αντιστάσεις (πτώση τάσης) ή εξωτερική τροφοδοσία (παραπάνω τάση)

- **Βαθμονόμηση (Calibration)**
  - Πολύ σημαντικό ρόλο παίζει η σωστή ρύθμιση των αισθητήρων σε σχέση με τις τιμές και τις ενδείξεις που επιστρέφουν
  - Απαιτούνται επομένως αρκετές δοκιμές έως ότου διεξαχθεί το επιθυμητό αποτέλεσμα

Δυνατότητες

- Συμπερασματικά, με τις δυνατότητες που παρέχει το Raspberry και σε συνδυασμό με το πλήθος των αισθητήρων που υπάρχουν στην αγορά, μπορούν να υλοποιηθούν πρωτότυπες και πολύ χρήσιμες κατασκευές - εφαρμογές στα πλαίσια της ρομποτικής αλλά και του αυτοματισμού

- Περισσότερα παραδείγματα θα ακολουθήσουν στα επόμενα μαθήματα



Βιβλιογραφία

[1] https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/

[2] https://fritzing.org/

[3] https://fritzing.org/projects/hc-sr04-project

[4] https://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/

[5] https://bigl.es/ds18b20-temperature-sensor-with-python-raspberry-pi/

[6] https://tutorials-raspberrypi.com/raspberry-pi-measure-humidity-temperature-dht11-dht22/

[7] https://www.instructables.com/Raspberry-Pi-GPIO-Circuits-Using-an-LDR-Analogue-S/