

Raspberry Pi- Προγραμματισμός LED

Ομάδα Μαθητών

:

Δραστηριότητα 1 -Τροφοδοσία LED μέσω του Raspberry Pi

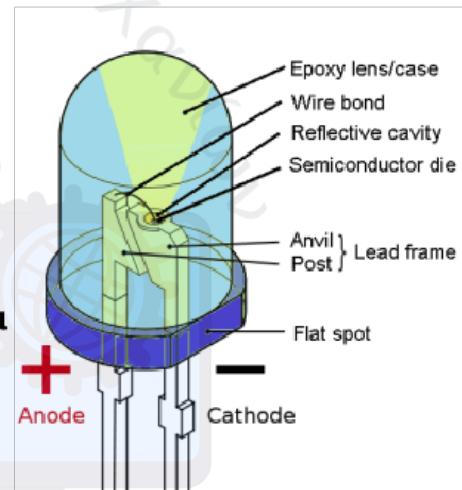
Υλικά που θα χρειαστείτε.

- 1) Raspberry Pi
- 2) Μία αντίσταση 270Ω έως 330Ω
- 3) 1 LED 5V
- 4) 1 breadboard
- 5) 2 καλώδια σύνδεσης (αρσενικό - θηλυκό)

Δίοδος Εκπομπής Φωτός (Light Emitting Diode) ή απλά LED

Το **LED** είναι μία ηλεκτρονική δίοδος από ημιαγώγιμα υλικά. Όταν περνάει ηλεκτρικό ρεύμα από δίοδο τύπου LED, αυτή εκπέμπει φως.

Όταν το LED τροφοδοτείται με ρεύμα **συνδέεται πάντα σε σειρά με μια αντίσταση**. Η αντίσταση περιορίζει το ηλεκτρικό ρεύμα και προστατεύει το LED από το να καεί.

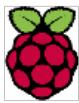


Εργασία με ηλεκτρονικές συσκευές - Στατικός Ηλεκτρισμός

Όλες οι ηλεκτρονικές συσκευές είναι ευπαθείς στο στατικό ηλεκτρισμό. Γι' αυτό το λόγο πρέπει πάντα να φροντίζουμε να μη φέρουμε πάνω μας στατικό φορτίο καθώς πιάνουμε τις πλακέτες.

Όταν δουλεύουμε με ηλεκτρονικά κυκλώματα για έξτρα προστασία καλό θα ήταν να είμαστε "γειωμένοι". Για να το κάνουμε αυτό πρέπει να κρατάμε κάποιο μεταλλικό αντικείμενο που καταλήγει στο έδαφος.

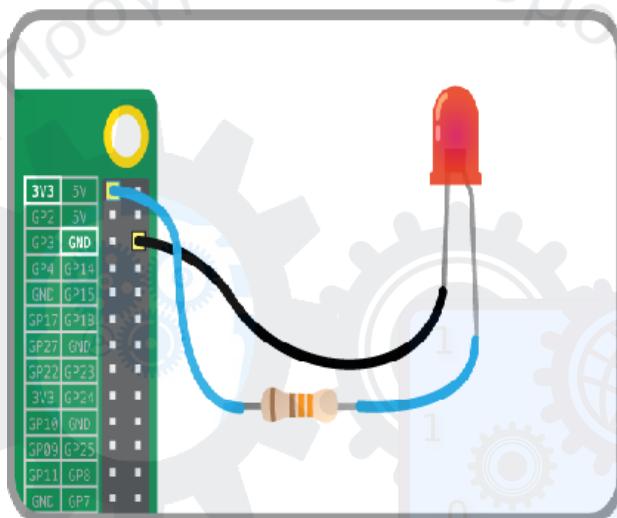
Μια πιο εύκολη λύση είναι τα ειδικά αντιστατικά βραχιόλια που συνδέουν τον καρπό μας με ένα μεταλλικό αντικείμενο (πχ το πόδι του πάγκου εργασίας).



Δραστηριότητα 1 - Τροφοδοσία LED μέσω του Raspberry Pi

Οδηγίες σύνδεσης LED με το Raspberry Pi

- Συνδέστε την κάθιδο (κοντό πόδι) του LED με τη γείωση (σύμφωνα με το διάγραμμα)
 - Συνδέστε την άνοδο (μακρύ πόδι) με το ένα άκρο της αντίστασης.
 - Συνδέστε το άλλο άκρο της αντίστασης με τον ακροδέκτη (σύμφωνα με το διάγραμμα).
- (Δεν έχει σημασία ποιο άκρο της αντίστασης συνδέετε στο LED και ποιο στο PIN).
- Τροφοδοτήστε με ρεύμα το Raspberry Pi.



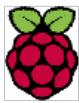
Τι παρατηρείτε-

Εξηγείστε;

.....
.....
.....
.....
.....
.....
.....

Τι μάθατε από την πραγματοποίηση αυτής της δραστηριότητας;

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....



Δραστηριότητα 2 - Προγραμματιζόμενο άναμα/σβήσιμο του LED

Στη Δραστηριότητα 1 χρησιμοποιήσαμε το Raspberry Pi, για να τροφοδοτήσουμε το LED με ρεύμα. Εφόσον όλα λειτουργούν κανονικά το λαμπάκι ανάβει και παραμένει συνεχώς αναμμένο. Στην παρούσα δραστηριότητα θα τροποποιήσουμε κατάλληλα τη συνδεσμολογία του προηγούμενου κυκλώματος, ώστε το άναμμα/σβήσιμο του Led να ελέγχεται από λογισμικό που θα γράψουμε χρησιμοποιώντας την Python, και εκτελείται από το Raspberry Pi.

Οδηγίες σύνδεσης LED με το Raspberry Pi

Για να πραγματοποιήσετε τη συνδεσμολογία σωστά συμβουλευτείτε το διάγραμμα <<2.Αριθμηση GPIO>> του φύλλου 1 “Ακροδέκτες Raspberry Pi”.

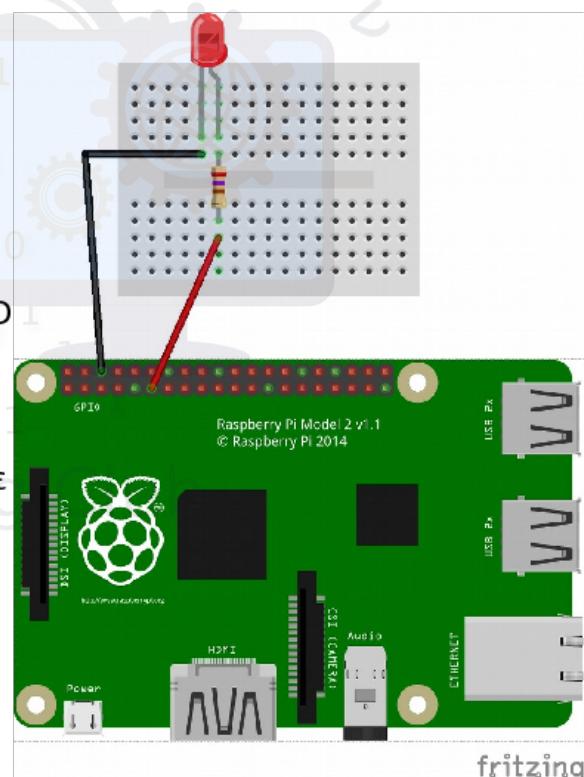
Τρόπος 1 -Αν έχετε μόλις ολοκληρώσει την δραστηριότητα 1

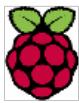
Αποσυνδέστε την τροφοδοσία του LED από τον ακροδέκτη του Raspberry Pi που παρέχει σταθερή τροφοδοσία 3.3Volt και συνδέστε το σε ένα ακροδέκτη με προγραμματιζόμενη έξοδο.

Σε αυτήν την κατητογορία ανήκουν όλοι οι κίτρινοι ακροδέκτρες. Στην δραστηριότητα αυτή επιλέγουμε τον **ακροδέκτη GPIO 17**.

Τρόπος 2: Αν ξεκινάτε τη σύνδεση από την αρχή.

- Συνδέστε την κάθοδο (κοντό πόδι) του LED με τη γείωση (μαύρο ακροδέκτη).
- Συνδέστε την άνοδο (μακρύ πόδι) με το ένα άκρο μιας αντίστασης.
- Συνδέστε το άλλο άκρο της αντίστασης με τον ακροδέκτη GPIO 17.
- Ελέγξτε προσεχτικά το κύκλωμα σας.





Δραστηριότητα 2 - Προγραμματιζόμενο άναμα/σβήσιμο του LED

- Για να γράψουμε το πρόγραμμα μας θα χρησιμοποιήσουμε το περιβάλλον προγραμματισμού **Geany**.
- Για το χειρισμό του LED θα αξιοποιήσουμε τη βιβλιοθήκη **gpiozero** που διαθέτει συναρτήσεις χειρισμού διαφόρων συσκευών όπως (LED, PWM LED, RGB LED , Buzzer , Μοτέ, Κουμπί, Αισθητήρες Κίνησης/Φωτός κλπ)

α) Δημιουργούμε ένα νέο αρχείο με όνομα **Led_on_off.py** και εισάγουμε τον παρακάτω κώδικα.

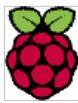
Led_on_off.py	Υποερώτημα γ
from gpiozero import LED from time import sleep led = LED(17) while True: led.on() sleep(1) led.off() sleep(1)	

Για να τρέξουμε το παραπάνω πρόγραμμα στο Raspberry Pi που αντιστοιχεί στην ομάδα σας θα πρέπει να συνδεθείτε σε αυτό απομακρυσμένα χρησιμοποιώντας το SSH, και να μεταφέρετε το αρχείο **Led_on_off.py**, στο Raspberry Pi, ακολουθώντας τις οδηγίες που σας δίνονται στο φύλλο 2: **Χρησιμοποιώντας το SSH για απομακρυσμένη σύνδεση στο Raspberry Pi**. Για να εκτελέσετε ένα πρόγραμμα γράφετε την παρακάτω εντολή στο τερματικό ssh.

python3 <όνομα αρχείου.py> (π.χ. python3 led_on_off.py)

β) Εκτελέστε τον παραπάνω κώδικα. Τι παρατηρείτε;

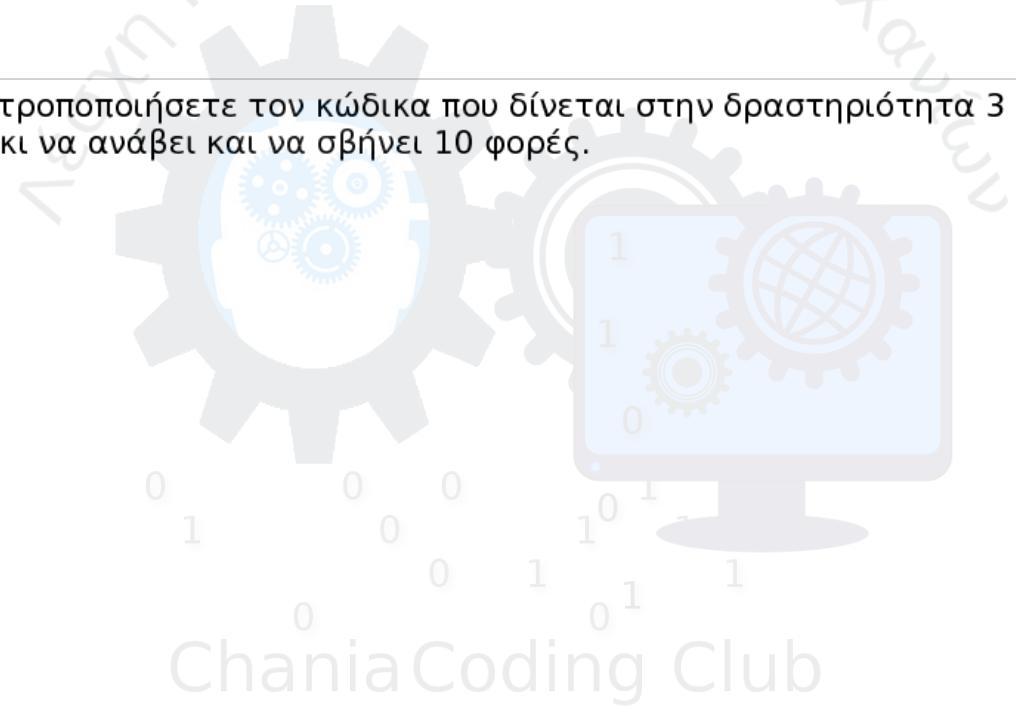
γ) Συμπληρώστε στο κουτάκι με τίτλο **Υποερώτημα γ** τροποποίηση του προγράμματος **Led_on_off.py** οπου το LED θα ανάβει για 5 δευτερόλεπτα και θα σβήνει για 2 δευτερόλεπτα.



Δραστηριότητα 3 - Ήρα για σκέψη και δημιουργία.

- α) Να φτιάξετε το διάγραμμα του κυκλώματος και τον αντίστοιχο κώδικα μέσω του οποίου θα ελέγχετε δύο LED. Αρχικά τα LED θα μπορούν να ανάβουν είτε ταυτόχρονα. Στη συνέχεια τροποποιείστε το πρόγραμμα σας ώστε τα LED να ανάβουν εναλλάξ.

β) Να τροποποιήσετε τον κώδικα που δίνεται στην δραστηριότητα 3 ώστε το λαμπάκι να ανάβει και να σβήνει 10 φορές.

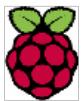


Πηγές:

GPIO Pin Numbering

<https://www.raspberrypi.org/learning/physical-computing-guide/worksheet/>





Λέσχη Προγραμματισμού Χανίων γλώσσα Python

Προγραμματισμός του Raspberry Pi με την

Σύνδεση **LED** στο RasberryPi:

<https://www.raspberrypi.org/learning/physical-computing-guide/connect-led/>

gpiozero homepage <https://gpiozero.readthedocs.io/en/v1.3.1/>

GPIO ZERO: A friendly python API for physical computing:

<https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/>

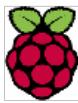
LED

http://www.garyfallidou.org/gr_simple_circuit_led.html

Σύνδεση LED



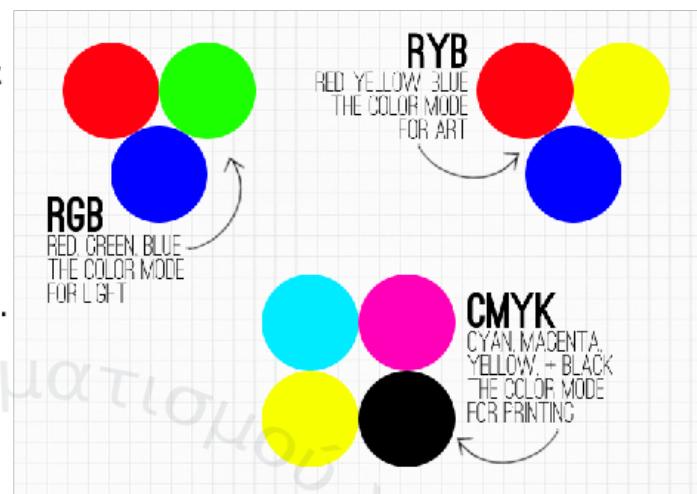
Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Παραγωγή Χρώματος

Βασικά και δευτερογενή χρώματα

Βασικά (ή πρωτογενή) χρώματα είναι τα χρώματα τα οποία όταν συνδυαστούν μεταξύ τους παράγουν τα υπόλοιπα χρώματα. Τα χρώματα που προκύπτουν από τον συνδυασμό των βασικών λέγονται **δευτερογενή**. Για την περιγραφή των χρωμάτων χρησιμοποιούμε τα χρωματικά μοντέλα ή **πρότυπα χρώματος**.



Πρότυπα Χρώματος

Στη ζωγραφική χρησιμοποιείται κυρίως το πρότυπο **RYB**, στις εκτυπώσεις το **CMYK**, ενώ το πρότυπο χρώματος **RGB** χρησιμοποιείται στις συσκευές οπτικής απεικόνισης (π.χ. οθόνες καθοδικού σωλήνα, υγρών κρυστάλλων ή πλάσματος).

Το πρότυπο χρώματος RGB

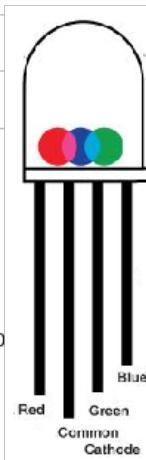
Το πρότυπο χρώματος **RGB** (Red, Green, Blue) έχει ως βασικά χρώματα το κόκκινο, το πράσινο και το μπλε. Το χρώμα κάθε εικονοστοιχείου παριστάνεται από μια τριάδα αριθμών (R,G,B) που δηλώνει πόσο κόκκινο, πράσινο και μπλέ περιέχει.



Τα περισσότερα προγράμματα και συσκευές (οθόνες, κάρτες γραφικών κλπ) χρησιμοποιούν για την αναπαράσταση κάθε χρώματος 1Byte (8bit). Έτσι μπορούμε να έχουμε 2^8 δηλαδή 256 διαφορετικές τιμές για κάθε βασικό χρώμα που κυμαίνται από 0 έως 255. Συνολικά μπορούμε να έχουμε $2^8 \times 2^8 \times 2^8 = 2^{24} = 2^4 \times 2^{20} = 16 = 16.777.216$ διαφορετικά χρώματα!

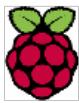
Τριάδες R,G,B

Χρώμα	R,G,B	Χρώμα	R.G.B
Κόκκινο- RED	255 ,0,0	κυανό-CYAN	0,255,255
Πράσινο-GREEN	0,255,0	ματζέντα-MAGENTA	
Μπλέ -BLUE	0,0,255	κίτρινο-YELLOW	



Έγχρωμη Δίοδος Εκπομπής Φωτός (RGB LED)





Η έχρωμη δίοδος εκπαμπής φωτός (RGB LED) περιέχει 3 διαφορετικές διόδους εκπομπής φωτός (3 led) κόκκινου, πράσινου και μπλε χρώματος μέσα στο ίδιο λαμπάκι. Το χρώμα που παράγεται εξαρτάται από το ποιες από τις τρεις διόδους <<ανάβουν>> κάθε φορά και με ποια ένταση.

Χρησιμοποιώντας το RGB LED - τεχνικά χαρακτηριστικά

Κάθε ηλεκτρονική συσκευή έχει συγκεκριμένα τεχνικά χαρακτηριστικά και προδιαγραφές λειτουργίας που πρέπει να λαμβάνουμε υπ' όψιν κατά τη σχεδίαση του ψηφιακού συστήματος που την ενσωματώνει. Για παράδειγμα για ένα RGB led θα πρέπει να γνωρίζουμε :

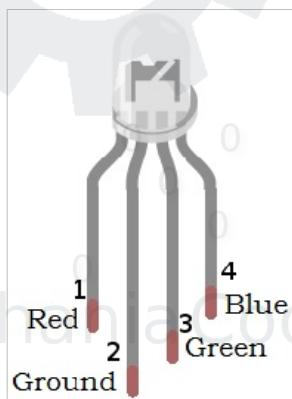
- α) τη λειτουργία του κάθε ακροδέκτη
- β) τις συνθήκες ομαλής λειτουργίας (π.χ. τάση, ένταση ηλεκτρικού ρεύματος κλπ).

Για την ομαλή λειτουργία του RGB LED θα πρέπει οι ακροδέκτες Red, Green, Blue να συνδέονται σε σειρά με την κατάλληλη αντίσταση ώστε το ρεύμα που διαρρέει το κύκλωμα να μην υπερβαίνει τις επιτρεπτές τιμές. Διαφορετικά μπορεί να έχουμε βλάβη στα εξαρτήματα που συνδέονται στο κύκλωμα (LED, raspberry Pi κλπ).

Σε αυτό το εργαστήριο θα χρησιμοποιήσουμε ένα τύπο RGB led που ονομάζονται led κοινής καθόδου, τα χαρακτηριστικά του οποίου φαίνονται στον παρακάτω πίνακα.

Ακροδέκτες RGB led κοινής καθόδου (common cathode RGB LED):

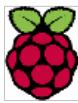
Τοποθετούμε το RGB LED ώστε ο μεγαλύτερος σε μήκος ακροδέκτης να είναι ο δεύτερος από αριστερά προς δεξιά.



Τοποθετούμε το RGB LED ώστε η επίπεδη πλευρά να είναι αριστερά.



Όταν το led τοποθετηθεί σωστά ο 1ος, 3ος και 4ος ακροδέκτης, από αριστερά προς τα δεξιά αντιστοιχούν στο Red, Green, Blue, ενώ ο 2ος συνδέεται στη γείωση (Ground ή GND).



Δραστηριότητα 1 - Προγραμματισμός RGB LED με το Raspberry Pi

Υλικά:

- 1 έγχρωμο RGB LED κοινής καθόδου
- 3 αντιστάσεις
- 4 καλώδια σύνδεσης (αρσενικό – θηλυκό)
- Raspberry Pi
- Breadboard

Σύνδεση RGB LED με το Raspberry Pi

α) Τοποθετήστε το LED σύμφωνα με τις οδηγίες τις σελίδας 1, έτσι ώστε να έχουμε την παρακάτω αντιστοιχία στους ακροδέκτες:

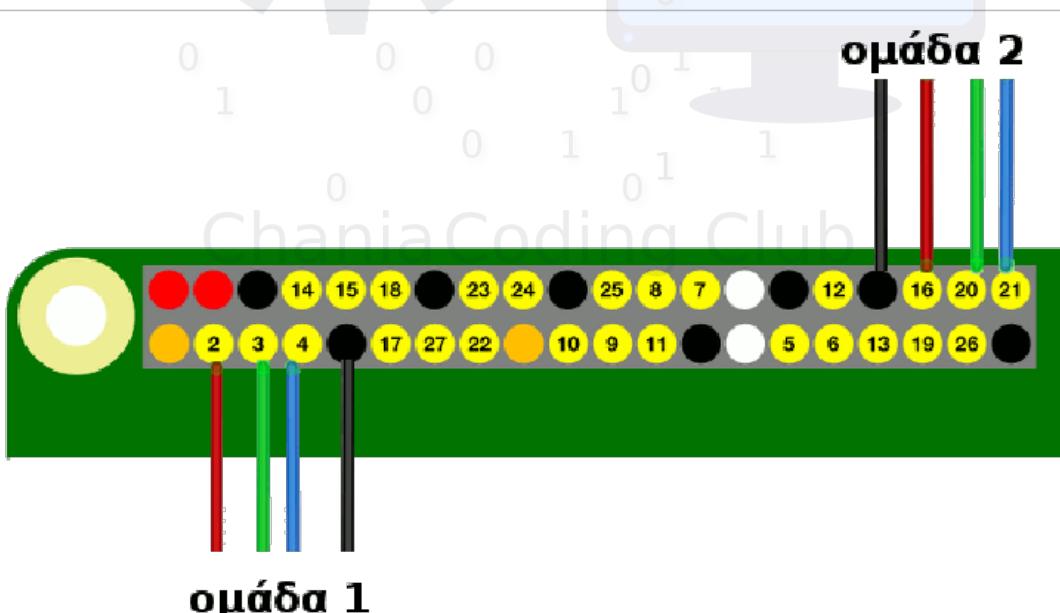
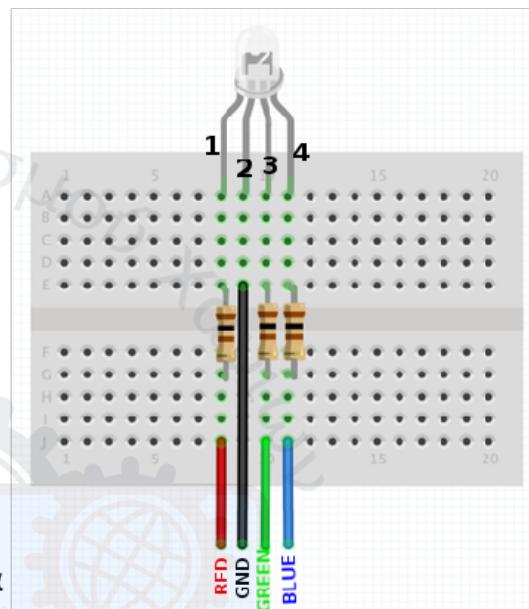
1 → Red , 2→GND, 3→Green, 4->Blue

β) Συνδέστε τον ακροδέκτη 2 με ένα καλώδιο.

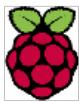
γ) Συνδέστε κάθε ένα από τους ακροδέκτες 1, 3, 4 σε σειρά μιά αντίσταση, όπως φαίνεται στο δίπλα σχήμα.

Ένα Raspberry Pi για δύο ομάδες

Σε αυτό το εργαστήριο κάθε Raspberry Pi θα χρησιμοποιείται ταυτόχρονα από δύο ομάδες. Κάθε ομάδα θα χρησιμοποιεί το δικό της σύνολο ακροδεκτών για να μπορούν να είναι ταυτόχρονα συνδεδεμένα και τα δύο κυκλώματα και να τρέχουν συγχρόνως τα προγράμματα και των δύο ομάδων.



Ομάδα 1 : Red→GPIO 2, Green→GPIO 3, Blue→GPIO 4
Ομάδα 2 : Red→GPIO 16, Green→GPIO 20, Blue →GPIO 21



Δραστηριότητα 1 - Προγραμματισμός RGB LED με το Raspberry Pi

Δημιουργούμε ένα πρόγραμμα αρχικά να δοκιμάσουμε τη λειτουργία του RGB led.

RGB_Led.py

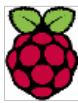
```
from gpiozero import RGBLED
from time import sleep
#ορίζουμε ποιοι ακροδέκτες ελέγχουν το κόκκινο, πράσινο, και μπλέ χρώμα
led = RGBLED(red=2, green=3, blue=4)    # μόνο για την ομάδα 1
led = RGBLED (red=16, green=20, blue=21)# μόνο για την ομάδα 2
print ("έλεγχος RGB led :έναρξη")
# καθαρό κόκκινο
led.red = 1
led.green=0
led.blue=0
sleep(3)
# καθαρό πράσινο
led.color = (0,1,0)
sleep(3)
# καθαρό μπλέ
led.color = (0,0,1)
sleep(3)
print ("έλεγχος RGB led : λήξη")
```

Εκτελέστε παραπάνω πρόγραμμα στο Raspberry Pi γράφοντας στο τερμοτικό::.

python3 RBG_Led.py

Τι παρατηρείτε;

.....
.....
.....
.....
.....
.....



Δραστηριότητα 2 - Τα χρώματα του ουράνιου τόξου

A. Να τροποποιήσετε το παραπάνω πρόγραμμα ώστε να παράγει τα χρώματα του ουράνιου τόξου με τη σειρά που φαίνονται στον παρακάτω πίνακα (από το κόκκινο έως το ιώδες).

Παρατήρηση: Για να ορίσουμε την ένταση με την οποία ανάβει το κάθενα από τα τρία led (R,G,B) χρησιμοποιούμε δεκαδικές τιμές μεταξύ 0 και 1.

π.χ.

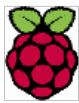
led.red = 0.6	Η εναλλακτικά
led.green = 0	Led.color=(0.6, 0, 0.2)
led.blue = 0.2	

B. Να συμπληρώσετε στον παρακάτω πίνακα τις ρυθμίσεις που τις τιμές (R.G.B) που χρησιμοποιήσατε κατά τον προγραμματισμό του RGB led για να επιτύχετε ένα ικανοποιητικό οπτικό αποτέλεσμα.

Μια προσέγγιση των χρωμάτων του ουράνιου τόξου	Ρυθμίσεις led		
	Red (0 έως 1)	Green(0 έως 1)	Blue(0 έως 1)
RGB(255,0,0)	1		
RGB(255,127,0)	1		
RGB(255,255,0)	0	1	
RGB(0,255,0)	0	1	
RGB(0,0,255)	0	0	1
RGB(75,0,130)	0	1	1
RGB(143,0,255)	0	1	1

Παρατηρήσεις:

.....
.....
.....
.....
.....
.....
.....
.....
.....



Πηγές:

Πρότυπο χρώματος RGB : <https://el.wikipedia.org/wiki/RGB>

GPIO Pin Numbering

<https://www.raspberrypi.org/learning/physical-computing-guide/worksheet/>

Σύνδεση LED στο RasberryPi:

<https://www.raspberrypi.org/learning/physical-computing-guide/connect-led/>

gpiozero homepage <https://gpiozero.readthedocs.io/en/v1.3.1/>

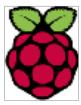
GPIO ZERO: A friendly python API for physical computing:

<https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/>

Προέλευση Εικόνων:

RGB, CMYK, RYB, <http://www.brandigirlblog.com/2013/02/rgb-vs-cmyk.html>

RGB model : <http://gentlelogic.blogspot.gr/2011/11/exploring-h264-part-1-color-models.html>



Αξιοποίηση έτοιμου κώδικα, μέσω των βιβλιοθηκών

Όταν γράφουμε ένα πρόγραμμα χρησιμοποιούμε συνήθως τις βασικές εντολές της αντίστοιχης γλώσσας προγραμματισμού. Όμως στην πράξη δεν είναι πάντα πρακτικό να δημιουργήσουμε ένα πρόγραμμα που εκτελεί σύνθετες λειτουργίες χρησιμοποιώντας μόνο τις βασικές εντολές.

Στην πραγματικότητα πριν επιλύσουμε προγραμματιστικά ένα πρόβλημα κάνουμε μια έρευνα για να δούμε αν κάποιοι άλλοι έχουν, με επιτυχία, επιλύσει το ίδιο πρόβλημα, ή ένα μέρος του προβλήματος πριν από μας. Η πρακτική της αξιοποίησης έτοιμου κώδικα είναι πολύ συνηθισμένη. Για το λόγο αυτό έχουν δημιουργηθεί συλλογές από προγράμματα που ονομάζονται **βιβλιοθήκες**. Κάθε βιβλιοθήκη περιέχει έτοιμα προγράμματα που μπορούμε να ενσωματώσουμε στα δικό μας. Ο προγραμματιστής ανάλογα με τις ανάγκες του κάθε φορά επιλέγει τις βιβλιοθήκες που χρειάζεται να χρησιμοποιήσει στο πρόγραμμα του.

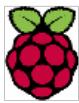
Για τις ανάγκες του προγράμματος χειρισμού του LED χρησιμοποιήσαμε δύο βιβλιοθήκες:

- Τη βιβλιοθήκη **gpiozero** που μας δίνει τη δυνατότητα να ελέγχουμε τη λειτουργία διαφόρων εξαρτημάτων που μπορούν να συνδεθούν στο Raspberry Pi (**LED**, **PWM LED**, **RGB LED**, **Buzzer**, **Μοτέρ**, **Κουμπί**, **Αισθητήρες Κίνησης/Φωτός ed**, μετατροπείς *analogue-to-digital*, κιτ ρομποτικής κ.α.). Αυτό γίνεται με απλό διαισθητικό τρόπο και πολύ λίγο κώδικα.
- Τη βιβλιοθήκη **time** που περιέχει έτοιμα υποπρογράμματα για τη διαχείριση του χρόνου. Από τη βιβλιοθήκη αυτή χρησιμοποιούμε την **sleep()**, που καθυστερεί την εκτέλεση του προγράμματος για όσα δευτερόλεπτα καθορίσουμε.

```
from gpiozero import LED
from time import sleep
led = LED(17)
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Για να χρησιμοποιήσουμε ένα στοιχείο μιας βιβλιοθήκης θα πρέπει πρώτα να το εισάγουμε. Αυτό στην Python χρησιμοποιώντας την εντολή: **from βιβλιοθήκη import στοιχείο**

Κώδικας Python	Λειτουργία
from gpiozero import LED	Από τη βιβλιοθήκη gpiozero εισάγω την κλάση LED
from time import sleep	Από τη βιβλιοθήκη time εισάγω τη συνάρτηση sleep
led = LED(17)	Δημιουργούμε ένα στιγμιότυπο της κλάσης LED , στο οποίο μπορούμε να αναφερόμαστε στο εξής χρησιμοποιώντας το όνομα led .
led.on(), led.off()	Μέθοδοι της κλάσης LED και αφορούν το στιγμιότυπο led π.χ. Η μέθοδος led.on() είναι μια συνάρτηση, έχει οριστεί μέσα στην κλάση LED, και την καλούμε για το στιγμιότυπο led .



Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

RaspberryPi- Προγραμματισμός με ήχο

Ομάδα

Μαθητών :

Θεωρητικό μέρος

Στο raspberry μπορούμε να χρησιμοποιήσουμε 2 τύπους ηχείων, τα χαρακτηριστικά των οποίων περιγράφονται παρακάτω.

Buzzer-βομβητής- πιεζοηλεκτρικό ηχείο

Δε χρησιμοποιούν μαγνήτη ή πηνίο φωνής αλλά κάποιο πιεζοηλεκτρικό υλικό, όπως π.χ. επιχρυσωμένο polymer film, κεραμικό ή συνθετικό υλικό.

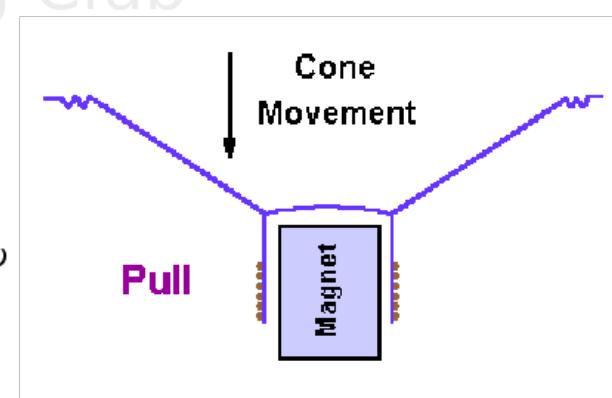
Είναι ένα ηχείο που χρησιμοποιεί το πιεζοηλεκτρικό φαινόμενο για την παραγωγή του ήχου. Η αρχική μηχανική κίνηση δημιουργείται με την εφαρμογή μιας τάσης σε ένα πιεζοηλεκτρικό υλικό, και αυτή η κίνηση μετατρέπεται σε ακουστικό ήχο χρησιμοποιώντας τυπικά Διαφράγματα και αντηχεία.

Ο πιεζοηλεκτρισμός είναι η ιδιότητα κάποιων υλικών (κυρίως κρυσταλλικών υλικών αλλά και μερικών κεραμικών υλικών) να παράγουν ηλεκτρική τάση όταν δέχονται κάποια μηχανική τάση/πίεση ή ταλάντωση. Το φαινόμενο μπορεί να εξηγηθεί ποιοτικά με τη μεταφορά ελεύθερων φορτίων στα άκρα του κρυσταλλικού πλέγματος. Επίσης, ο όρος περιλαμβάνει και το αντίστροφο φαινόμενο, κατά το οποίο το υλικό παραμορφώνεται, όταν βρεθεί κάτω από ηλεκτρική τάση. Ο πιεζοηλεκτρισμός ανακαλύφθηκε από τον Πιερ Κιουρί το 1880. Οφείλεται σε κρυσταλλικές ασυμμετρίες και αποτελεί φαινόμενο πρώτης τάξης.

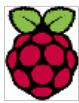


Ηλεκτροδυναμικά ηχεία:

Είναι τα πιο διαδεδομένα ηχεία. Αποτελούνται από ένα (ή δύο) πηνίο φωνής το οποίο είναι κεντραρισμένο στο διάκενο μεταξύ μαγνήτη και οπλισμού. Πάνω στο πηνίο είναι προσκολλημένος ο κώνος, ο οποίος «μεταφράζει» με την παλινδρομική κίνησή του το ηλεκτρικό ρεύμα σε ακουστική ενέργεια.



Για να μπορέσουμε να παράγουμε τον ήχο που επιθυμούμε στην άσκηση μας, θα χρησιμοποιήσουμε ηλεκτροδυναμικά ηχεία.

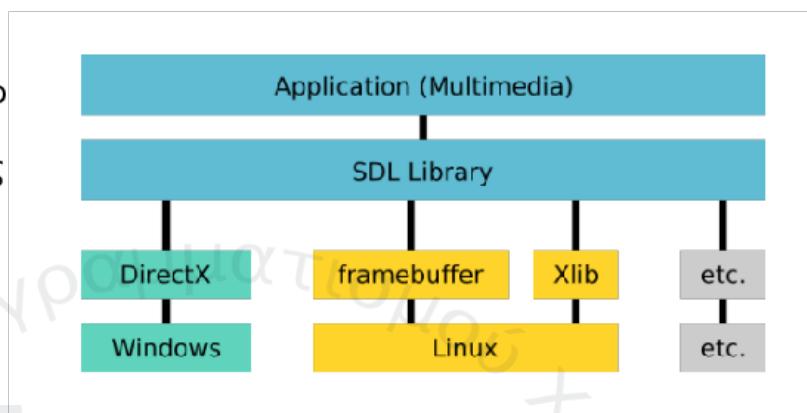


Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Τι είναι η Pygame

Η Pygame είναι μια επέκταση που παρέχει πρόσβαση στην βιβλιοθήκη SDL και τα βοηθήματα της, όπου SDL (Simple DirectMedia Layer) είναι μια πολυμεσική βιβλιοθήκη σχεδιασμένη να παρέχει πρόσβαση χαμηλού επιπέδου σε ήχο, πληκτρολόγιο, ποντίκι, joystick, υλικό 3D μέσω OpenGL και 2D βίντεο με συμβατότητα πολλών πλατφορμών.

Ως προς το SDL λοιπόν, το Pygame λειτουργεί αφαιρετικά προσφέροντας πόρους του συστήματος που θα απαιτούνται στην δημιουργία αλληλεπίδρασης χρήστη με γραφικά.



Η επέκταση OS

Η επέκταση OS της Python παρέχει έναν τρόπο να ενσωματώνουμε εντολές του λειτουργικού συστήματος μέσα στο πρόγραμμά μας. Η επέκταση είναι ανεξάρτητη λειτουργικού συστήματος. Μπορεί να λειτουργήσει σε όλα τα γνωστά λειτουργικά όπως Windows, Mac ή Linux με μικρές αλλαγές κάθε φορά.

TTS (Text-To-Speech)

Σε αυτή την άσκηση χρησιμοποιήσαμε μία υπηρεσία όπου μπορεί και κάνει ανάγνωση του κειμένου που έχουμε γράψει. Η υπηρεσία αυτή είναι online και μπορείτε να την βρείτε στις παρακάτω ηλεκτρονικές διευθύνσεις.

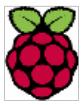
<https://www.innoetics.com/>

http://www.oddcast.com/home/demos/tts/tts_example.php?siterepal

Αφού γράψαμε το κείμενο και έγινε η εκφώνηση, το μετατρέψαμε σε ένα αρχείο ήχου με επέκταση ogg. Η επέκταση pygame μπορεί να αναπαράγει αρχεία ήχου τύπου ogg ή wav. Το ίδιο κάναμε και με τις 10 ερωτήσεις που χρησιμοποιούμε στο συγκεκριμένο φύλλο εργασίας, όπως παρουσιάζετε στον παρακάτω πίνακα.

Ερώτηση 1	1.ogg	Ερώτηση 6	6.ogg
Ερώτηση 2	2.ogg	Ερώτηση 7	7.ogg
Ερώτηση 3	3.ogg	Ερώτηση 8	8.ogg
Ερώτηση 4	4.ogg	Ερώτηση 9	9.ogg
Ερώτηση 5	5.ogg	Ερώτηση 10	10.ogg





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Πρακτικό μέρος

Σύνδεση

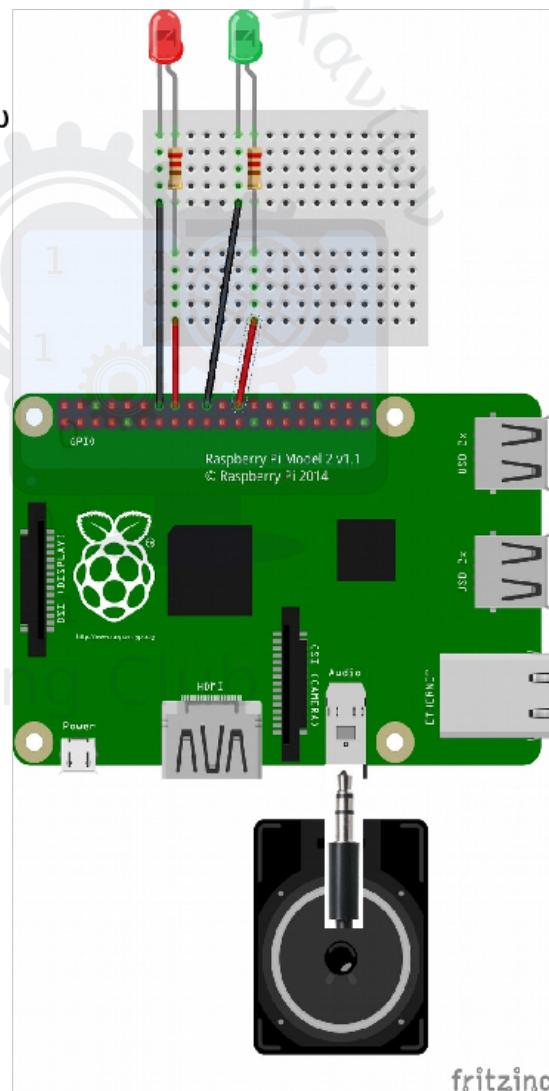
Υλικά που θα χρειαστείτε.

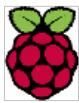
- 1) Raspberry pi
- 2) 2 αντίστάσεις 220Ω έως 330Ω
- 3) 2 Led 5V (κόκκινο, πράσινο)
- 4) 1 breadboard
- 5) 4 καλώδια σύνδεσης (αρσενικό - θηλυκό)
- 6) 1 Ήχειο με σύνδεση 3.5mm jack

Οδηγίες σύνδεσης

Ανά 2 οι ομάδες θα συνεργαστούν και θα κατασκευάσουν από κοινού την παρακάτω συνδεσμολογία.

Θα συνδέσετε το κόκκινο led στο gpio 23 και το πράσινο στο gpio 8. Στη συνέχεια θα συνδέσετε τις αντίστοιχες γειώσεις. Τέλος θα συνδέσετε το βύσμα του ηχείου στη μοναδική υποδοχή που ταιριάζει και είναι ίδια με αυτή των ακουστικών που έχουν τα κινητά σας τηλέφωνα.





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Δραστηριότητα 1 - Προγραμματισμός Δημιουργία παιχνιδιού ερωτήσεων

Αρχικά θα εισάγουμε τις βιβλιοθήκες που θα χρειαστούμε.

```
import pygame
from time import sleep
import os
```

Η παρακάτω εντολή θα προσπαθήσει να αρχικοποιήσει όλα τα δομικά στοιχεία από το οποίο αποτελείται η επέκταση pygame. Πιθανόν να μην χρειαστούμε όλα αυτά τα δομικά στοιχεία. Μπορούμε να αρχικοποιήσουμε συγκεκριμένα δομικά στοιχεία με το «χέρι». Π.χ. αν θέλουμε να αρχικοποιήσουμε τις γραμματοσειρές γράφουμε την εντολή pygame.font.init()

```
pygame.init()
```

Ας καθαρίζουμε την οθόνη μας κάθε φορά που το «τρέχουμε»

```
os.system('clear')
```

Πάμε να τυπώσουμε την πρώτη ερώτηση

```
print ("Ποιο από τα παρακάτω ΔΕΝ ήταν παιδί του Ουρανού;")
print("1.Γαία")
print("2.Εγκέλαδος")
print("3.Κρόνος")
print("4.Ωκεανός")
```

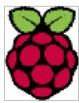
Με την παρακάτω εντολή φορτώνουμε το αρχείο ήχου με όνομα 1.ogg που περιέχει την πρώτη ερώτηση.

```
pygame.mixer.music.load("1.ogg")
```

Στη συνέχεια παίζει το αρχείο αυτό από την έξοδο που έχουμε συνδέσει το αρχείο του raspberry.

```
pygame.mixer.music.play()
```





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Ας αναλύσουμε λίγο την παρακάτω εντολή.

Γράφοντας `input("Δώσε Απάντηση:")` ζητάμε από τον χρήστη του προγράμματος να μας δώσει από το πληκτρολόγιο την απάντηση της ερώτησης. Βάζοντας το παραπάνω μέσα στη συνάρτηση `int()` προσπαθεί να μετατρέψει αυτό που θα δώσει ο χρήστης σε ακέραιο αριθμό. Τέλος βάζει την απάντηση σε μια μεταβλητή που την ονομάζουμε `apantisi`.

```
apantisi=int(input("Δώσε Απάντηση:"))
```

Ήρθε η ώρα να ελέγξουμε αν η ερώτηση απαντήθηκε σωστά.

Η απάντηση στην ερώτηση είναι το 1, οπότε ελέγχουμε με την εντολή `if` αν η μεταβλητή `apantisi` έχει την τιμή 1. Αν έχει την τιμή 1 τότε φορτώνει το αρχείο `y.ogg` και μετά το παίζει στα ηχεία περιμένοντας για 2 δευτερόλεπτα μέχρι να ολοκληρωθεί.

Άν ο χρήστης δεν δώσει 1, αλλά οποιοδήποτε άλλο αριθμό τότε εκτελεί τις εντολές κάτω από το `else`, οπότε φορτώνει το αρχείο `n.ogg` και μετά το παίζει στα ηχεία περιμένοντας για 2 δευτερόλεπτα μέχρι να ολοκληρωθεί, μετά φορτώνει το αρχείο `ap1.ogg` και το παίζει στα ηχεία περιμένοντας για 3 δευτερόλεπτα μέχρι να ολοκληρωθεί.

Το αρχείο `y.ogg` περιέχει μήνυμα οτι η απάντηση είναι σωστή.

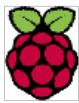
Το αρχείο `n.ogg` περιέχει μήνυμα οτι η απάντηση είναι λάθος.

Το αρχείο `ap1.ogg` περιέχει μήνυμα οτι η απάντηση είναι η το 1.

Αντίστοιχα υπάρχουν τα αρχεία `ap2.ogg`, `ap3.ogg`, `ap4.ogg` που περιέχουν εκφωνήσεις για τις αντίστοιχες απαντήσεις.

```
if (apantisi == 1):
    pygame.mixer.music.load("y.ogg")
    pygame.mixer.music.play()
    sleep(2)
else:
    pygame.mixer.music.load("n.ogg")
    pygame.mixer.music.play()
    sleep(2)
    pygame.mixer.music.load("ap1.ogg")
    pygame.mixer.music.play()
    sleep(3)
```





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Η τελική μορφή του προγράμματος σας πρέπει να είναι η παρακάτω.

```
import pygame
from time import sleep
import os

pygame.init()

os.system('clear')

print ("Ποιο από τα παρακάτω ΔΕΝ ήταν παιδί του Ουρανού;")
print("1.Γαία")
print("2.Εγκέλαδος")
print("3.Κρόνος")
print("4.Ωκεανός")

pygame.mixer.music.load("1.ogg")
pygame.mixer.music.play()

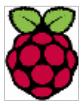
apantisi=int(input("Δώσε Απάντηση:"))

if (apantisi == 1):
    pygame.mixer.music.load("y.ogg")
    pygame.mixer.music.play()
    sleep(2)
else:
    pygame.mixer.music.load("n.ogg")
    pygame.mixer.music.play()
    sleep(2)
    pygame.mixer.music.load("ap1.ogg")
    pygame.mixer.music.play()
    sleep(3)
```

Δραστηριότητα 2

Προσπαθήστε να προσθέσετε κώδικα στο πρόγραμμα σας ώστε να ανάβει το κόκκινο led αν ο χρήστης απαντήσει λάθος ή το πράσινο αν απαντήσει σωστά. Στη συνέχεια γράψτε στο παρακάτω πλαίσιο τις εντολές που χρησιμοποιήσατε.



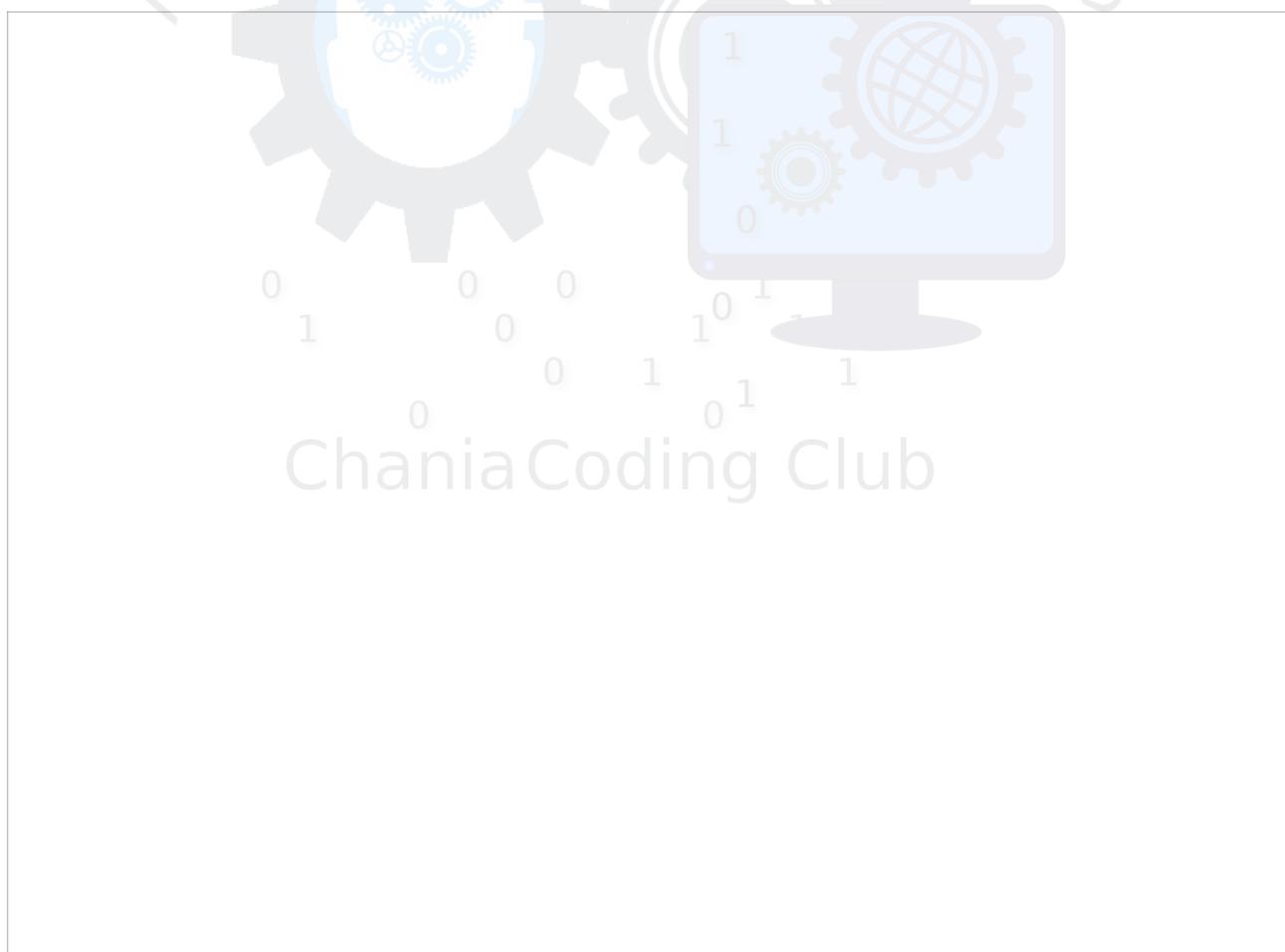


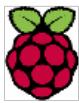
Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python



Δραστηριότητα 3

Τώρα προσπαθήστε να συνεχίσετε το quiz προσθέτοντας την δεύτερη ερώτηση στο πρόγραμμά σας. Όλες οι ερωτήσεις βρίσκονται στις τελευταίες σελίδες του φύλλου εργασίας. Οι σωστές απαντήσεις είναι με πιο έντονο χρώμα. Στη συνέχεια γράψτε στο παρακάτω πλαίσιο τις εντολές που χρησιμοποιήσατε.





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Δραστηριότητα 4

Δημιουργήστε μια μεταβλητή όπου θα κρατάει το σκορ. Κάθε σωστή απάντηση δίνει 20 βαθμούς. Κάθε λάθος απάντηση αφαιρεί 10. Στο τέλος των ερωτήσεων θα παρουσιάζετε το σκορ που έπιασε ο παίκτης.

Δραστηριότητα 5

Τώρα που γίνατε expert συνεχίστε φτιάχνοντας και τις υπόλοιπες ερωτήσεις.

ΕΡΩΤΗΣΕΙΣ QUIZ

Ερώτηση 1

Ποιο από τα παρακάτω ΔΕΝ ήταν παιδί του Ουρανού;

1. Γαία

- 2. Εγκέλαδος
- 3. Κρόνος
- 4. Ωκεανός

Ερώτηση 2

Ποια ήταν η μητέρα του Δία;

1. Ρέα

- 2. Νεφέλη
- 3. Θέμιδα
- 4. Φοίβη

Ερώτηση 3

Ποια ήταν η θεά της μαγείας;

1. Εκάτη

- 2. Περσεφόνη
- 3. Ήρα
- 4. Ήβη

Ερώτηση 4

Με ποιους τάχθηκαν οι Κύκλωπες στην Τιτανομαχία (και πολέμησαν στο πλευρό τους);

1. Με τους Ολύμπιους Θεούς

- 2. Με τους Τιτάνες

Ερώτηση 5

Τι κρατάει ο Ερμής στις απεικονίσεις του;

- 1. Σπαθί
- 2. Γραφίδα
- 3. Ράβδο
- 4. Ακόντιο

Ερώτηση 6

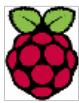
Σε τι μεταμορφώθηκε ο Δίας για να ζευγαρώσει με τη Δανάη;

- 1. Άτλαντας

2. Προμηθέας

- 3. Δίας





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

4.Ηφαιστος

Ερώτηση 7

Ποιο ήταν το μόνο "δώρο" που δεν ξέφυγε όταν η Πανδώρα άνοιξε το περίφημο κουτί της; (που κατά τη μυθολογία ήταν πιθάρι)

1.Η Ειρήνη

2.Η Ελπίδα

3.Η Πίστη

4.Η Αγάπη

Ερώτηση 8

Ποιο από τα παρακάτω κατορθώματα του Ηρακλή ΔΕΝ ανήκει στους 12 άθλους;

1.Η αρπαγή των Βοδιών του Γηρυόνη

2.Ο φόνος του Ερυμάνθιου Κάπρου

3.Ο φόνος του Κενταύρου Νέστου

4.Η αρπαγή του Ελαφιού της Κερύνειας

Ερώτηση 9

Ποια ήταν η δεύτερη σύζυγος του Θησέα; (πρώτη ήταν η αμαζόνα Ιππολύτη)

1.Αριάδνη

2.Φαίδρα

Ερώτηση 10

Ποιος ήταν ο προορισμός της Αργοναυτικής εκστρατείας; (η τοποθεσία στην οποία βρισκόταν το χρυσόμαλλο δέρας που ήθελε να πάρει ο Ιάσονας)

1.Άργος

2.Σαμοθράκη

3.Κολχίδα

4.Ιωλκός

Πηγές:

GPIO Pin Numbering

<https://www.raspberrypi.org/learning/physical-computing-guide/worksheet/>

Σύνδεση Led στο RasberryPi:

<https://www.raspberrypi.org/learning/physical-computing-guide/connect-led/>

gpiozero homepage <https://gpiozero.readthedocs.io/en/v1.3.1/>

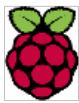
GPIO ZERO: A friendly python API for physical computing:

<https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/>

<http://www.pygame.org/lofi.html>



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

RaspberryPi- Προγραμματισμός με Υπερηχητικό Αισθητήρα Απόστασης

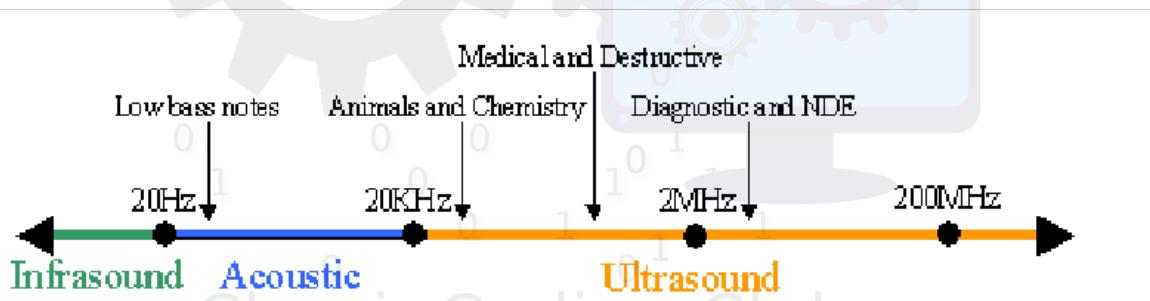
Ομάδα Μαθητών :

Θεωρητικό μέρος

Ο αισθητήρας υπερήχων μετράει αποστάσεις σε εκατοστά και ίντσες. Ο υπερηχητικός αισθητήρας χρησιμοποιεί την **ίδια** επιστημονική αρχή με τα **ραντάρ**, τα **σόναρ** αλλά και τις **νυχτερίδες**: μετρά την απόσταση με τον υπολογισμό του χρόνου που παίρνει ένα κύμα για να χτυπήσει ένα αντικείμενο και να επιστρέψει - ακριβώς όπως μια ηχώ. Τα μεγάλου μεγέθους αντικείμενα με τις σκληρές επιφάνειες επιστρέφουν τις καλύτερες αναγνώσεις. Τα αντικείμενα φτιαγμένα από μαλακό ύφασμα ή τα κυρτά [όπως μια σφαίρα] ή τα πολύ λεπτά ή μικρά μπορεί να είναι δύσκολο να ανιχνευθούν από τον αισθητήρα.

Υπέρηχος

Σαν **υπέρηχο** καθορίζουμε εκείνο το κύμα το οποίο βρίσκεται πάνω από την μέγιστη συχνότητα που μπορεί να ακούσει το ανθρώπινο αυτή. Παρόλο πάντως που εμείς δεν τους ακούμε κάποια ζώα μπορούν και να τους ακούν αλλά και να τους χρησιμοποιούν. Χαρακτηριστικά παραδείγματα που μας το δείχνουν αυτό είναι η κίνηση των νυχτερίδων και η σφυρίχτρα που χρησιμοποιείται για τους σκύλους.

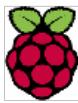


Τρόπος υπολογισμού απόστασης:

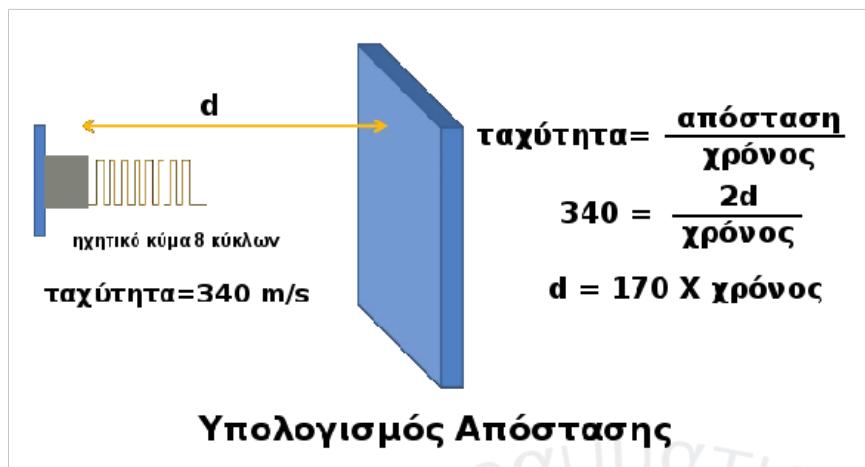
Στον αέρα, ο ήχος ταξιδεύει με ταχύτητα 343 μέτρα το δευτερόλεπτο. Ένας υπερηχητικός αισθητήρας απόστασης στέλνει παλμούς υπερήχων που είναι μη αντιληπτοί από τους ανθρώπους, και ανιχνεύει την ηχώ που στέλνεται πίσω όταν ο ήχος αντανακλάται από ένα κοντινό αντικείμενο. Στην συνέχεια χρησιμοποιεί την ταχύτητα του ήχου για να υπολογίσει την απόσταση από τα αντικείμενο

Τύπος υπολογισμού απόστασης:





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python



Υπολογισμός Απόστασης

Εφαρμογές υπερηχητικών αισθητήρων απόστασης

Εφαρμογές των υπερηχητικών αισθητήρων απόστασης θα βρούμε σε ένα μεγάλο εύρος τεχνολογιών από την μέτρηση της διεύθυνσης και της ταχύτητας του ανέμου έως και την απεικονιστική ιατρική. Οι υπέρηχοι αποτελούν μια από τις σημαντικότερες μη καταστροφικές μεθόδους ελέγχου υλικών.

Ο υπερηχητικός αισθητήρας επιτρέπει σε ένα ρομπότ να δει και να ανιχνεύσει τα αντικείμενα, να αποφύγει εμπόδια, να μετρά αποστάσεις και να ανιχνεύει μετακινήσεις.

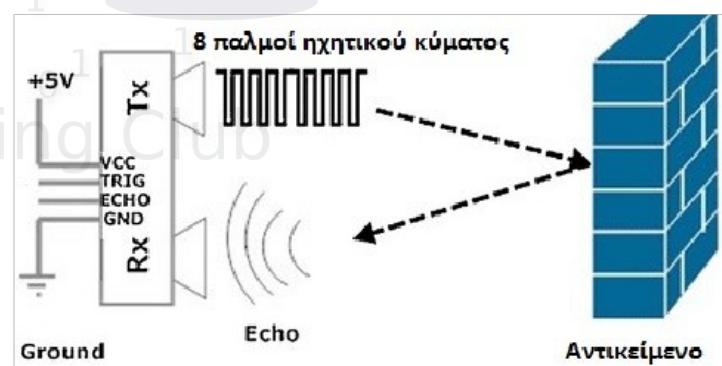
Αισθητήρας απόστασης HC-SR04

Στο εργαστήριό μας θα χρησιμοποιήσουμε τον υπερηχητικό αισθητήρα απόστασης HC-SR04 που έχει τα παρακάτω χαρακτηριστικά:



Τάση λειτουργίας: 5V
 Εμβέλεια: 2 cm - 500 cm
 Ανάλυση - Ακρίβεια: 0,3 cm
 Συχνότητα (υπερήχου): 40 kHz
 Cycle Period: 50 ms

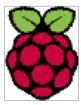
τέσσερεις ακροδέκτες:
VCC δέχεται 5 Volt
GND συνδέεται σε γείωση
TRIG (είσοδος) που ενεργοποιεί τον αισθητήρα
ECHO (έξοδος) που επιστρέφει το σήμα (ηχώ)



Πρακτικό μέρος

Σύνδεση





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

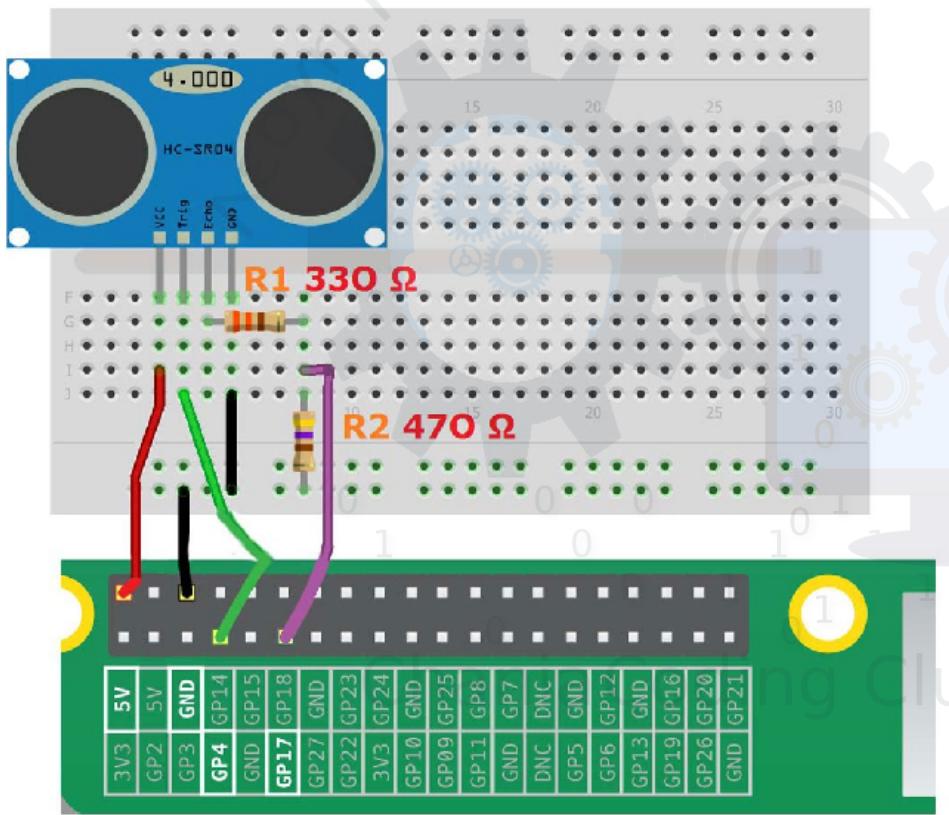
Υλικά που θα χρειαστείτε.

- 1) Raspberry pi
- 2) 2 αντιστάσεις 330Ω και 470Ω
- 3) 1 breadboard
- 4) 4 καλώδια σύνδεσης (αρσενικό – θηλυκό)
- 5) 2 καλώδια σύνδεσης (αρσενικό – αρσενικό)

Οδηγίες σύνδεσης

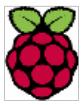
Το κύκλωμα συνδέεται σε δύο GPIO ακροδέκτες (ένα για την **ηχώ**, και ένα για **πυροδότηση (trigger)**), τον ακροδέκτη γείωσης και ένα ακροδέκτη 5V. Θα χρησιμοποιήσουμε ένα ζευγάρι αντιστάσεις (330 Ω και 470 Ω) ως **διαιρέτη τάσης**.

Γιατί διαιρέτη τάσης;

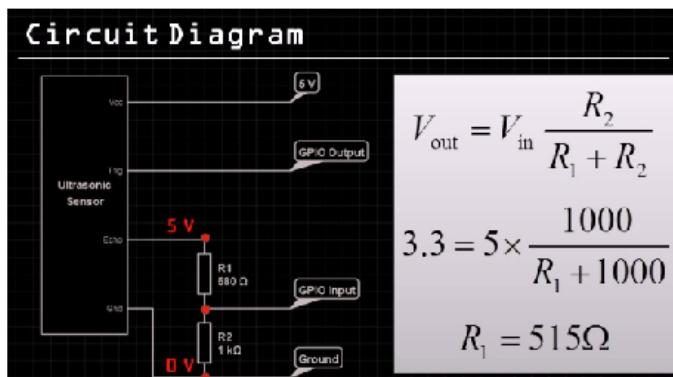


Για να **προστατεύσουμε** το **Raspberry** μας καθώς οι **ακροδέκτες GPIO** μπορούν να δεχτούν μέχρι το μέγιστο **3.3 V** Η διαιρέση τάσης μας επιτρέπει να έχουμε την επιθυμητά χαμηλή τάση στο ακροδέκτη που συνδέεται με το σήμα **Echo** (που μπορεί να δώσει **Υψηλή τιμή σήματος 5 Volt**). Με τον διαιρέτη εξασφαλίζουμε ότι το σήμα **Echo** που στέλνει ο αισθητήρας στο **Raspberry** θα έχει τάση έως **3.3 V** και δεν θα προκαλείται ζημιά στα **GPIO pins**





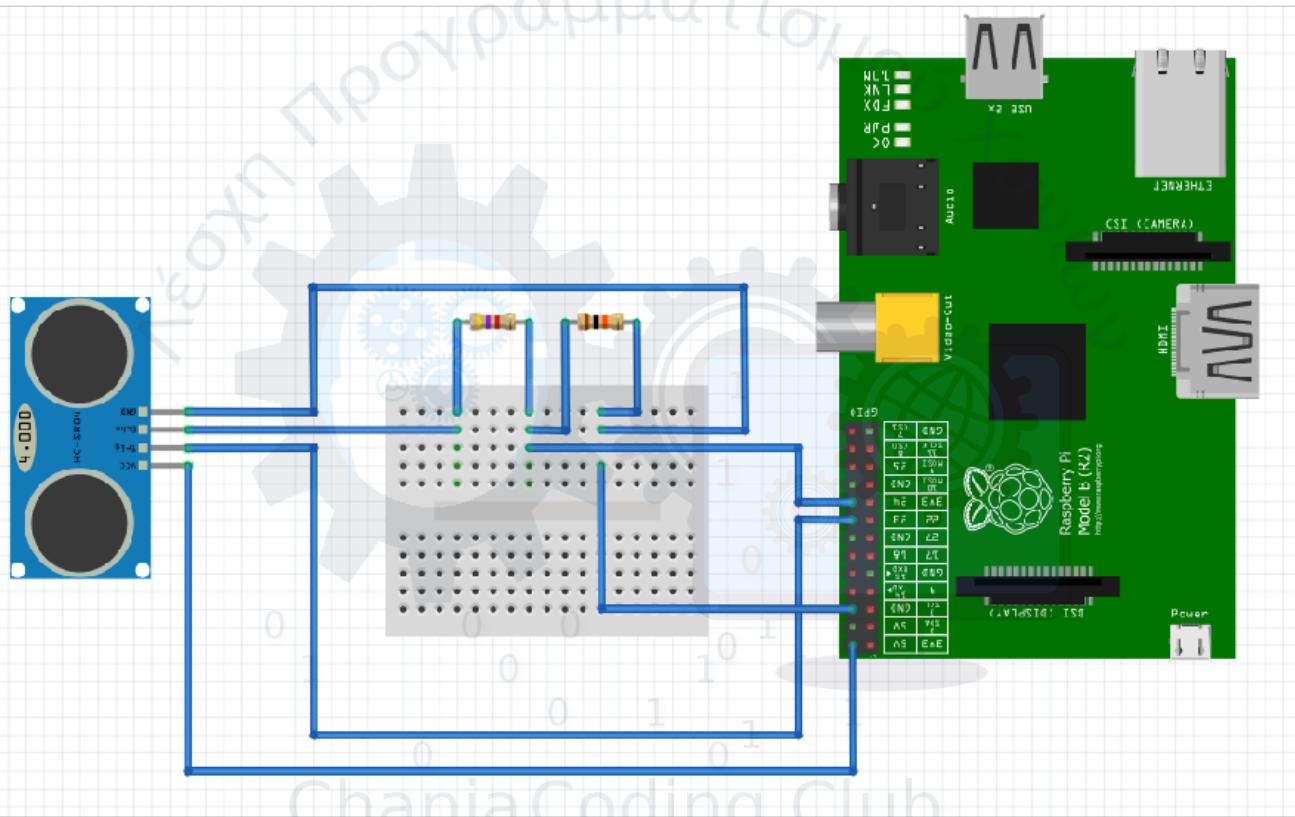
Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry πί με τη γλώσσα Python



Ο παρακάτω τύπος δείχνει τον τρόπο επιλογής των αντιστάσεων Ανά 2 οι ομάδες θα συνεργαστούν και θα κατασκευάσουν από κοινού την παραπάνω συνδεσμολογία.

Δραστηριότητα 1 -

Προγραμματισμός/Εμφάνιση απόστασης αντικειμένου από αισθητήρα



Για να χρησιμοποιήσουμε τον υπερηχητικό αισθητήρα απόστασης στην Python θα πρέπει να εισάγουμε από την βιβλιοθήκη **gpiozero** την κλάση **DistanceSensor**

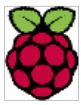
Αρχικά θα εισάγουμε τις βιβλιοθήκες που θα χρειαστούμε.

```
from gpiozero import DistanceSensor  
from time import sleep  
import os
```

Δημιουργούμε ένα στιγμιότυπο του **DistanceSensor** συνδέοντας τον ακροδέκτη **echo** με το **gpio 17** και τον ακροδέκτη **trigger** με το **gpio 4**.

```
ultrasonic = DistanceSensor(echo=17, trigger=4)
```





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Ζητάμε στην συνέχεια να εμφανιστεί η απόσταση από τον αισθητήρα

ultrasonic.distance

οπότε και πρέπει να δούμε την απόσταση σε μέτρα από το κοντινότερο στον αισθητήρα αντικείμενο

Για να βλέπουμε συνεχώς την απόσταση, ενώ κυματίζουμε το χέρι μας μπροστά από τον αισθητήρα που θα διαβάζει την απόσταση θα πρέπει να χρησιμοποιήσουμε τον γνωστό μας **ατέρμονα βρόγχο**

```
while True:  
    print(ultrasonic.distance)
```

Η τιμή που θα εμφανίζεται θα γίνεται διαρκώς μικρότερη όσο πλησιάζουμε στον αισθητήρα Για έξοδο από την επανάληψη πρέπει να πατήσουμε **Ctrl+C**.

Η τελική μορφή του προγράμματος σας πρέπει να είναι η παρακάτω:

```
from gpiozero import DistanceSensor  
from time import sleep  
import os  
  
ultrasonic = DistanceSensor(echo=17, trigger=4)  
  
while True:  
    print(ultrasonic.distance)
```

Δραστηριότητα 1β - Προγραμματισμός Εμφάνιση απόστασης αντικειμένου από αισθητήρα

α) Κάντε τον αισθητήρα να διατηρεί το μήνυμα απόστασης όταν αυτό εμφανίζεται για 3 δευτερόλεπτα, και στην συνέχεια να καθαρίζει την οθόνη.

Υπόδειξη: Χρησιμοποιήστε την μέθοδο **sleep** και την κλήση στην εντολή **'clear'** -του λειτουργικού συστήματος {Συμβουλευτείτε την Δραστηριότητα 1 του προηγούμενου φύλλου έργου - "Προγραμματισμός με ήχο" }

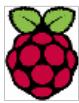
β) Κάντε κάθε φορά να εμφανίζεται ο αύξων αριθμός της μέτρησης πριν την απόσταση.

π.χ. την 5η φορά να εμφανίζεται το μήνυμα **5: απόσταση 0,72 μέτρα**

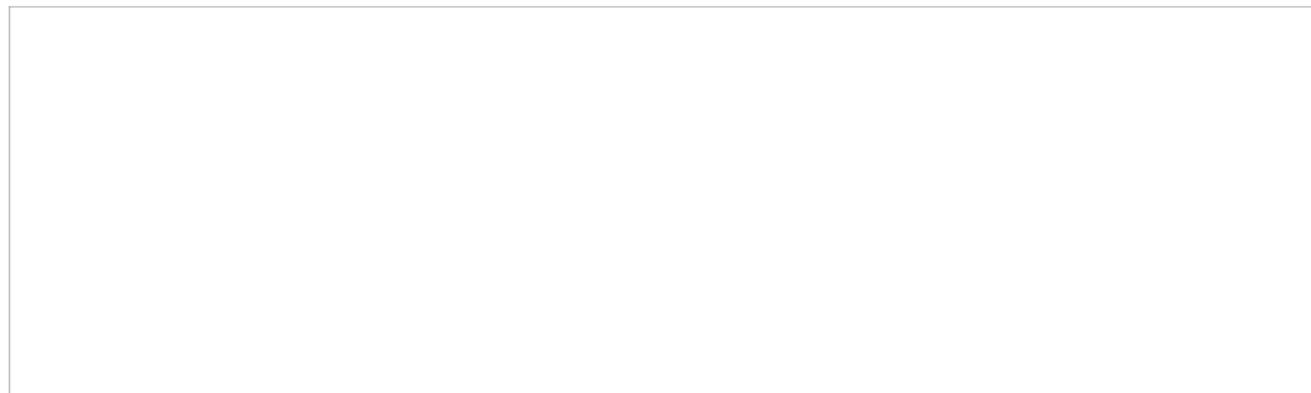
Υπόδειξη: Χρησιμοποιήστε ένα μετρητή το οποίο σε κάθε επανάληψη θα αυξάνεται κατά 1

Στη συνέχεια γράψτε στο παρακάτω πλαίσιο τις εντολές που χρησιμοποιήσατε.

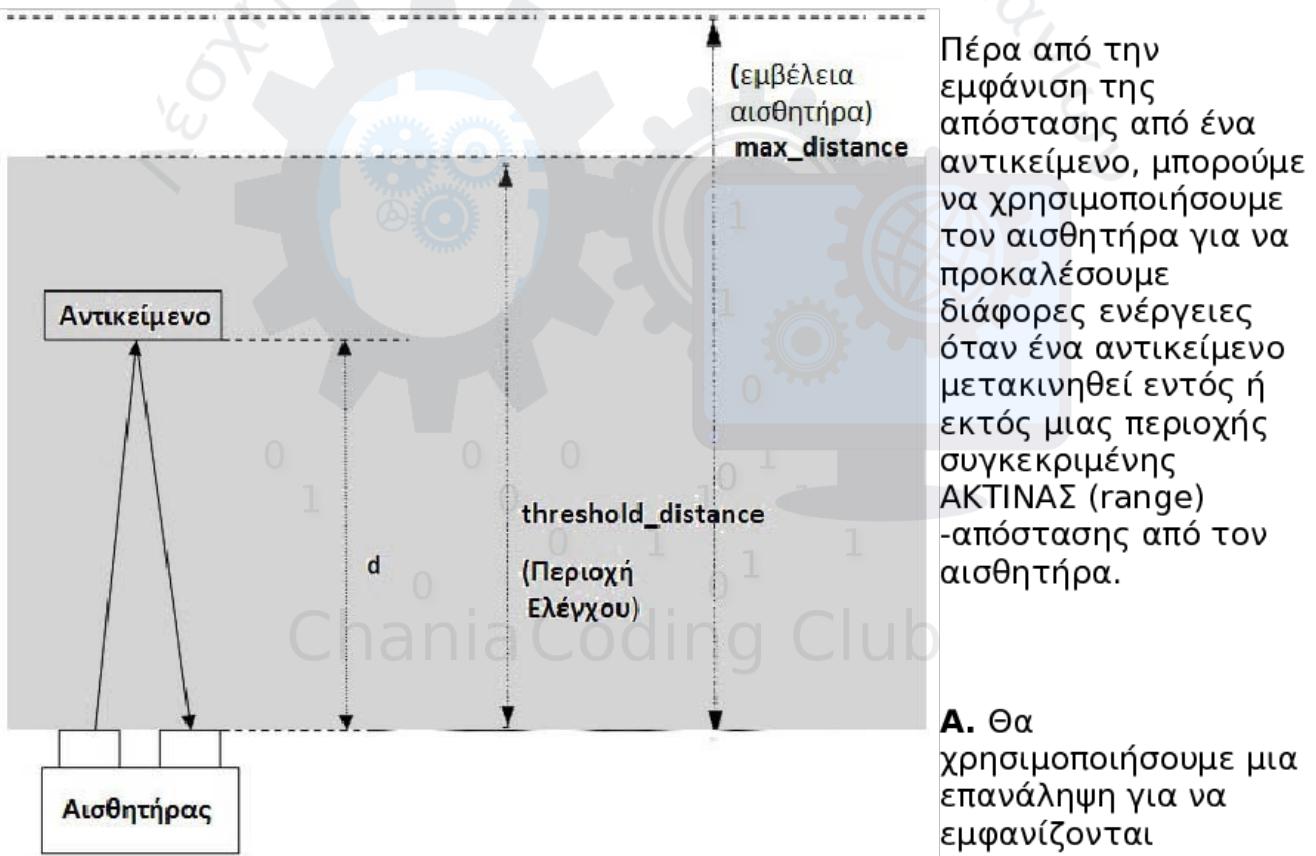




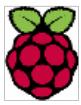
Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python



Δραστηριότητα 2 - Προγραμματισμός Εμφάνιση μηνύματος όταν μετακινηθούμε μέσα ή έξω από περιοχή ΣΥΓΚΕΚΡΙΜΕΝΟΥ εύρους-απόστασης από τον αισθητήρα



μόλις ένα αντικείμενο βρεθεί μέσα ή έξω από μια περιοχή συγκεκριμένου πλάτους – απόστασης (**περιοχή ελέγχου**) από τον αισθητήρα. Αρχικά θα κάνουμε το πρόγραμμα να “σταματήσει και να περιμένει (halt)” μέχρι την στιγμή που ένα αντικείμενο θα βρεθεί μέσα στην περιοχή - ζώνη συγκεκριμένου πλάτους γύρω από τον τον αισθητήρα, μέχρι δηλαδή να “ενεργοποιηθεί” η εντολή



Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

ultrasonic.wait_for_in_range()

Στην συνέχεια, όταν δηλαδή κάτι ανιχνευτεί μέσα στην συγκεκριμένη περιοχή, θα εμφανίσουμε το μήνυμα “**Εντός περιοχής - Σε έπιασα !!!**”

Ομοίως θα κάνουμε το πρόγραμμα να “σταματήσει και να περιμένει (halt)” μέχρι την στιγμή που το αντικείμενο θα μετακινηθεί και θα βρεθεί “έξω” από την συγκεκριμένη προσδιορισμένη εμβέλεια του αισθητήρα, δηλαδή μέχρι να “ενεργοποιηθεί” η εντολή

ultrasonic.wait_for_out_of_range()

Αμέσως μετά θα εμφανίσουμε το μήνυμα “**Εκτός περιοχής - Σε έχασα :-)**” Οι συναρτήσεις **wait_for** είναι συναρτήσεις “που παγώνουν” ένα πρόγραμμα μέχρι να συμβεί το γεγονός που θα περιμένουν.

Η τελική μορφή του προγράμματος σας πρέπει να είναι η παρακάτω:

```
from gpiozero import DistanceSensor  
ultrasonic = DistanceSensor(echo=17, trigger=4)
```

while True:

```
    ultrasonic.wait_for_in_range()  
    print(" Εντός περιοχής - Σε έπιασα, Καλωσόρισες !!! ")  
    ultrasonic.wait_for_out_of_range()  
    print("Εκτός περιοχής - Σε έχασα, Αντίο !!! ")
```

Ο αισθητήρας απόστασης έχει προκαθορισμένο εύρος - ακτίνα περιοχής τα 0,3 μέτρα.

Μπορείτε να **αλλάξετε** και το **εύρος - ακτίνα περιοχής** του αισθητήρα, αλλάζοντας την ιδιότητά του threshold_distance, με την παρακάτω εντολή, όταν αυτός **αρχικοποιείται**:

```
ultrasonic = DistanceSensor(echo=17, trigger=4, threshold_distance=0.5)
```

ή αφού έχει δημιουργηθεί με την εντολή

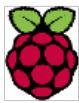
```
ultrasonic.threshold_distance = 0.5
```

Το εύρος περιοχής του αισθητήρα μπορεί να είναι ένας πραγματικός αριθμός. Στο παραπάνω παράδειγμα έγινε 0.5 μέτρα.

Συναρτήσεις - Υποπρογράμματα

Μία συνάρτηση (υποπρόγραμμα) είναι ένα ανεξάρτητο τμήμα κώδικα, ένα





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

αυτόνομο “εξάρτημα” που επιτελεί μια συγκεκριμένη λειτουργία. Ο ορισμός μιας συνάρτησης ξεκινά με τη λέξη **def** και ακολουθείται από το όνομα της συνάρτησης και τις παραμέτρους της, μέσα σε παρενθέσεις. Οι εντολές μετά την πρώτη γραμμή είναι στοιχισμένες δεξιότερα. Η στοίχιση αυτή υποδηλώνει ότι οι εντολές αυτές “ανήκουν” στη συνάρτηση και θα εκτελεστούν όταν αυτή κληθεί.

Παρακάτω δίνονται δύο συναρτήσεις, μια με όνομα **welcome**, και μια με όνομα **goodbye**

def welcome():

```
print(" Εντός περιοχής - Σε έπιασα, Καλωσόρισες !!! ")
```

def goodbye():

```
print("Εκτός περιοχής - Σε έχασα, Αντίο !!! ")
```

Μπορούμε να ορίσουμε όταν ένα αντικείμενα μπαίνει στην εμβέλεια του αισθητήρα να εκτελείται η συνάρτηση **welcome**, και όταν βγαίνει να εκτελείται η συνάρτηση **goodbye** με τις παρακάτω εντολές

```
ultrasonic.when_in_range = welcome  
ultrasonic.when_out_of_range = goodbye
```

Δραστηριότητα 2β - Προγραμματισμός

Ρύθμιση εύρους - απόστασης από αισθητήρα, εμφάνιση μηνύματος ελέγχου

Αλλάξετε το εύρος περιοχής - απόσταση από τον αισθητήρα σε **0,6 μέτρα**. Ξαναγράψετε το παραπάνω πρόγραμμα χρησιμοποιώντας τις συναρτήσεις **welcome** και **goodbye**.

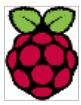
Συμπληρώσετε στο παρακάτω πλαίσιο τις εντολές που χρησιμοποιήσατε.

Chania Coding Club

Δραστηριότητα 2γ - Έλεγχος περιοχής - Ρύθμιση εμβέλειας αισθητήρα



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

Ο αισθητήρας σταματά να λειτουργεί στο 1 μέτρο. Μπορείτε να αλλάξετε την μέγιστη απόσταση - εμβέλεια λειτουργίας (πχ. σε 2 μέτρα) είτε κατά την αρχικοποίηση του αισθητήρα με την εντολή

```
ultrasonic = DistanceSensor(echo=17, trigger=4, max_distance=2)
```

είτε μετά την αρχικοποίηση με την εντολή

```
ultrasonic.max_distance = 2
```

Συμπληρώστε στο παρακάτω πλαίσιο μια μόνο εντολή που θα ρυθμίζει η περιοχή ελέγχου του αισθητήρα να έχει ακτίνα - εύρος 0.15 μέτρα και η εμβέλεια λειτουργίας του αισθητήρα να είναι τα 2.5 μέτρα.

Δραστηριότητα 3 - Παιχνίδι υπολογισμού απόστασης

Δημιουργήστε παιχνίδι το οποίο θα εμφανίζει ένα τυχαίο αριθμό από το 0,1 έως τα 0.8 μέτρα και θα ζητά από τον χρήστη να τοποθετήσει το χέρι του στην απόσταση που του ζητείται. Στον παίκτη θα δίνεται ένα χρονικό περιθώριο 5 δευτερολέπτων για να τοποθετήσει το χέρι του.

Όταν ο χρήστης καταφέρει να προσεγγίσει την απόσταση με απόκλιση 0,05 μέτρα θα εμφανίζεται το μήνυμα **μπράβο**, διαφορετικά θα εμφανίζεται το μήνυμα **απέτυχες**.

Το πρόγραμμα θα εκτελείται διαρκώς και θα διακόπτεται με πάτημα **Ctrl + C**.

Ορίστε ως **εμβέλεια** - δηλαδή μέγιστη απόσταση λειτουργίας του αισθητήρα τα 1,5 μέτρα.

Οδηγίες για την υλοποίηση του παιχνιδιού

Για να παραχθεί ένας τυχαίος αριθμός από το 0.1 έως το 0.8, αρκεί πρώτα να παράγετε ένα τυχαίο ακέραιο αριθμό από το 10 έως το 80 και το αποτέλεσμα να το διαιρέσετε με το 100

Για να παράγετε ένα τυχαίο αριθμό πρέπει να εισάγετε την συνάρτηση random με την εντολή

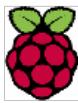
```
import random
```

Για να παράγετε ένα τυχαίο αριθμό από 10 έως 80 αρκεί να εισάγετε την εντολή

```
tyxaios = random.randint(10, 80)
```

και για να τον μετατρέψετε σε αριθμό από 0.1 έως 0.8 χρησιμοποιήστε την εντολή





Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

apostasi= tyxaios/100

Το πρόγραμμα επομένως θα πρέπει να συμπεριφέρεται όπως ο παρακάτω αλγόριθμος:

Πρώτα θα αρχικοποιεί τον αισθητήρα απόστασης

Διαρκώς θα εκτελούνται τα παρακάτω:

Θα δημιουργείται τυχαίος αριθμός από 0.1 εως 0.8

Θα διαβάζεται η απόσταση του χεριού

και θα συγκρίνεται με την τυχαία απόσταση

Εαν το **απόλυτο της διαφοράς τους** { ***abs(απόσταση_από_αισθητήρα - apostasi)*** }

είναι ΜΙΚΡΟΤΕΡΟ από 0.05

Θα εμφανίζεται μήνυμα **μπράβο**

διαφορετικά το μήνυμα **απέτυχες**

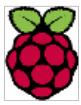
Συμπληρώστε το πρόγραμμα στο παρακάτω πλαίσιο

β) Τροποποιήστε το πρόγραμμα ώστε να παίζεται το αρχείο **bravo.ogg** κάθε φορά που ο παίκτης πετυχαίνει την απόσταση, και το **apotyxia.ogg** όταν αποτυχαίνει

γ) Τροποποιήστε το πρόγραμμα ώστε να εμφανίζει και σκορ στον χρήστη. Κάθε φορά που ο χρήστης πετυχαίνει το σκορ να αυξάνεται κατά 1 και όταν αποτυχαίνει κατά 0.5

Στις 10 αποτυχημένες προσπάθειες να τερματίζεται το πρόγραμμα

Συμπληρώστε το πρόγραμμα στο παρακάτω πλαίσιο



Λέξη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

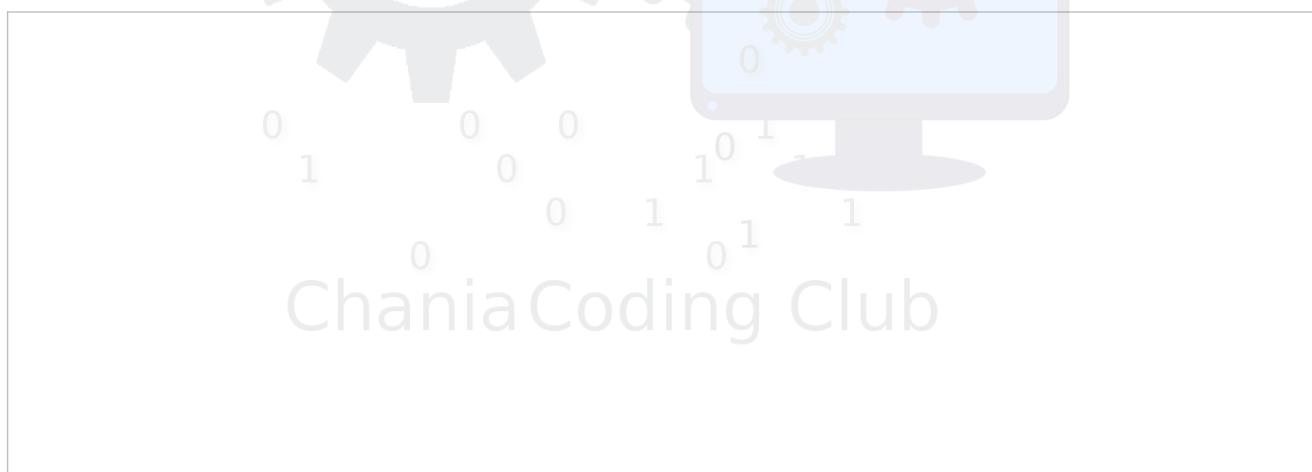


Δραστηριότητα 4

Δημιουργήστε πρόγραμμα το οποίο θα κάνει τα παρακάτω. Όταν ένα αντικείμενο βρίσκεται στην περιοχή ελέγχου του αισθητήρα θα ανάβει ένα Led το οποίο η ομάδα 1 θα έχει συνδέσει στο GPIO pin 7 και η ομάδα 2 θα έχει συνδέσει στο GPIO pin 11.

Όταν το αντικείμενο απομακρύνεται το Led θα σβήνει
Οι εργασίες ανάμματος και σβησίματος του Led μπορούν να γίνονται και μέσω συναρτήσεων όπως η **welcome** και η **goodbye**

Συμπληρώστε το πρόγραμμα στο παρακάτω πλαίσιο



Δραστηριότητα 5

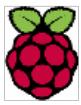
Δημιουργήστε πρόγραμμα το οποίο θα χρησιμοποιεί ένα RGB Led και όσο ένα αντικείμενο πλησιάζει θα ενισχύεται το κόκκινο, ενώ όσο απομακρύνεται θα ενισχύεται το κίτρινο.

Πηγές:

GPIO Pin Numbering



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων – Προγραμματισμός Raspberry pi με τη γλώσσα Python

<https://www.raspberrypi.org/learning/physical-computing-guide/worksheet/>
Σύνδεση Led στο RasberryPi:

<https://www.raspberrypi.org/learning/physical-computing-guide/connect-led/>

Αισθητήρας Απόστασης στο RasberryPi:

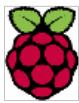
<https://www.raspberrypi.org/learning/physical-computing-with-python/distance/>
gpiozero homepage <https://gpiozero.readthedocs.io/en/v1.3.1/>

GPIO ZERO: A friendly python API for physical computing:

<https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/>



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Φωτοαντιστάσεις - Lasers

Θεωρητικό μέρος

Φωτοαντίσταση

Η λειτουργία της φωτοαντίστασης βασίζεται στη μεταβολή της τιμής της, ανάλογα με την ένταση του φωτός που πέφτει πάνω της. Σε απόλυτο σκοτάδι η αντίσταση είναι μέγιστη.

Εφαρμόζοντας όμως τάση στα άκρα της, εμφανίζεται ένα ρεύμα που οφείλεται κυρίως σε θερμική λειτουργία (ρεύμα σκότους).

Με την αύξηση της έντασης του φωτός, η αντίσταση του υλικού μικραίνει και επιτρέπει τη δημιουργία μεγάλων ρευμάτων.



Πυκνωτής

Σε πολλές από τις συσκευές που χρησιμοποιούμε στην καθημερινή ζωή, όπως τα στερεοφωνικά συγκροτήματα, οι τηλεοράσεις, οι τηλεφωνικές συσκευές, οι ηλεκτρονικοί υπολογιστές, υπάρχει η ανάγκη να αποθηκεύεται κάποια ποσότητα ηλεκτρικού φορτίου για ορισμένο χρονικό διάστημα και να χρησιμοποιείται την κατάλληλη χρονική στιγμή.

Η διάταξη με την οποία επιτυγχάνεται η αποθήκευση του ηλεκτρικού φορτίου είναι ο **πυκνωτής**.

Ο πυκνωτής είναι μία συσκευή που χρησιμεύει ως αποθήκη ηλεκτρικού φορτίου και επομένως ηλεκτρικής ενέργειας.



Αποτελείται από δύο αγωγούς που διαχωρίζονται από ένα μονωτικό υλικό.

Αποτελείται από ένα γυάλινο δοχείο το οποίο έχει καλυφθεί εσωτερικά και εξωτερικά με λεπτά φύλλα μετάλλου. Ένα κατακόρυφο μεταλλικό στέλεχος που περνά από το στόμιο του δοχείου, έχει στο επάνω μέρος του ένα μεταλλικό σφαιρίδιο και είναι μονωμένο με κατάλληλο πώμα. Το κάτω άκρο του στελέχους μέσω μιας μεταλλικής αλυσίδας έρχεται σε επαφή με το εσωτερικό φύλλο του μετάλλου-αγωγού.

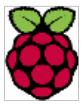
Οι δύο αγωγοί (φύλλα μετάλλου) εσωτερικός και εξωτερικός αποτελούν τον πυκνωτή. Αν με μία φορτισμένη ράβδο φορτίσουμε τον εσωτερικό αγωγό αρνητικά, τότε τα ηλεκτρόνια του εξωτερικού αγωγού απωθούμενα διαφεύγουν μέσω της γείωσης και ο αγωγός φορτίζεται θετικά. Διακόπτοντας την επαφή με την ράβδο και τη γείωση έχουμε ένα φορτισμένο πυκνωτή.

Χωρητικότητα πυκνωτή

Αν φορτίσουμε διαδοχικά έναν πυκνωτή, με φορτία Q, 2Q, 3Q κ.λπ. αποδεικνύεται, ότι η τάση του γίνεται αντίστοιχα V, 2V, 3V κ.λπ. Επομένως, το φορτίο και η τάση ενός πυκνωτή είναι μεγέθη ανάλογα. Το πηλίκο τους είναι χαρακτηριστικό μέγεθος του πυκνωτή, ονομάζεται, χωρητικότητα του πυκνωτή και συμβολίζεται με το γράμμα C.

Χωρητικότητα C ενός πυκνωτή ονομάζεται το μονόμετρο φυσικό μέγεθος που είναι ίσο με το πηλίκο του ηλεκτρικού φορτίου Q του πυκνωτή, προς την τάση V του πυκνωτή.

$$C = \frac{Q}{V}$$



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Μονάδα χωρητικότητας στο S.I. είναι το 1F

$$1\text{Farad} = \frac{1\text{Coulomb}}{1\text{Volt}}$$

Η χωρητικότητα ενός πυκνωτή είναι ένα πολύ χρήσιμο μέγεθος, γιατί μας πληροφορεί για το φορτίο που μπορεί να αποθηκευτεί ανά μονάδα τάσης μεταξύ των οπλισμών του.

Laser

Ο όρος λέιζερ ή λέηζερ προέρχεται από το αγγλικό ακρωνύμιο Laser: (Light Amplification by Stimulated Emission of Radiation) που αποδίδεται στα ελληνικά ως ενίσχυση φωτός με εξαναγκασμένη εκπομπή ακτινοβολίας και καλύπτει τόσο τις συσκευές που την παράγουν όσο και την αντίστοιχη ακτινοβολία.

Τα λέιζερ παράγουν συμφασικό, μονοχρωματικό φως (δηλαδή φως με συγκεκριμένο μήκος κύματος-χρώμα), το οποίο διαδίδεται σε μία συγκεκριμένη κατεύθυνση, σχηματίζοντας στενές δέσμες. Αντίθετα, οι συνηθισμένες πηγές φωτός, όπως οι λαμπτήρες πυρακτώσεως, παράγουν μη-σύμφωνο φως προς όλες τις διευθύνσεις και, επιπλέον, έχουν μεγάλο φασματικό εύρος.

Η λειτουργία των λέιζερ ερμηνεύεται από τη θεωρία της κβαντικής μηχανικής και της θερμοδυναμικής. Πολλά υλικά με τα απαραίτητα χαρακτηριστικά για να αποτελέσουν ενεργό υλικό των λέιζερ έχουν βρεθεί, με αποτέλεσμα τη δημιουργία πολλών τύπων λέιζερ με διαφορετικά χαρακτηριστικά, που χρησιμοποιούνται σε μεγάλο εύρος εφαρμογών.

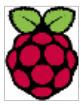
Η εφεύρεση των λέιζερ στηρίχθηκε στην κατασκευή των μέίζερ στη δεκαετία του 1950. Το πρώτο λέιζερ κατασκευάστηκε το 1960, από τότε όμως τα λέιζερ βρήκαν εφαρμογή στις θετικές επιστήμες, στη βιομηχανία, στην ιατρική, και στην ηλεκτρονική.



Προσοχή!!

Τα πράγματα είναι απλά για το λέιζερ. Μην το κοιτάτε απευθείας, μην σημαδεύετε με αυτό άλλα άτομα, μην σημαδεύετε καθρέφτες ή άλλες αντανακλαστικές επιφάνειες, και βεβαίως μην σημαδεύετε αυτοκίνητα, σκάφη ή αεροπλάνα. Αυτό ισχύει όποια ισχύ κι αν αναγράφει, ακόμα κι αν αυτή βρίσκεται κάτω από το θεωρούμενο ως όριο ασφαλείας των 5 milliWatts. Αν υπάρχει τραυματισμός, δεν είναι πολλά αυτά που μπορείτε να κάνετε. Να θυμάστε επίσης ότι η ακτίνα τους παραμένει ισχυρή και λαμπερή για πολλά μέτρα ή και χιλιόμετρα ακόμα από την πηγή της, επομένως ο κίνδυνος είναι πάντοτε υπαρκτός. Αν το λέιζερ που προμηθευτήκατε δεν φέρει επισήμανση για την ισχύ του, θεωρήστε το εξ ορισμού επικίνδυνο.





Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Πρακτικό μέρος Σύνδεση

Υλικά που θα χρειαστούμε

1 Raspberry pi

1 LED

1 αντίσταση 220 ohm

2 Breadboard

1 LDR - φωτοαντίσταση

1 Πυκνωτή 1 μ F

1 KY-008 Laser Transmitter Module

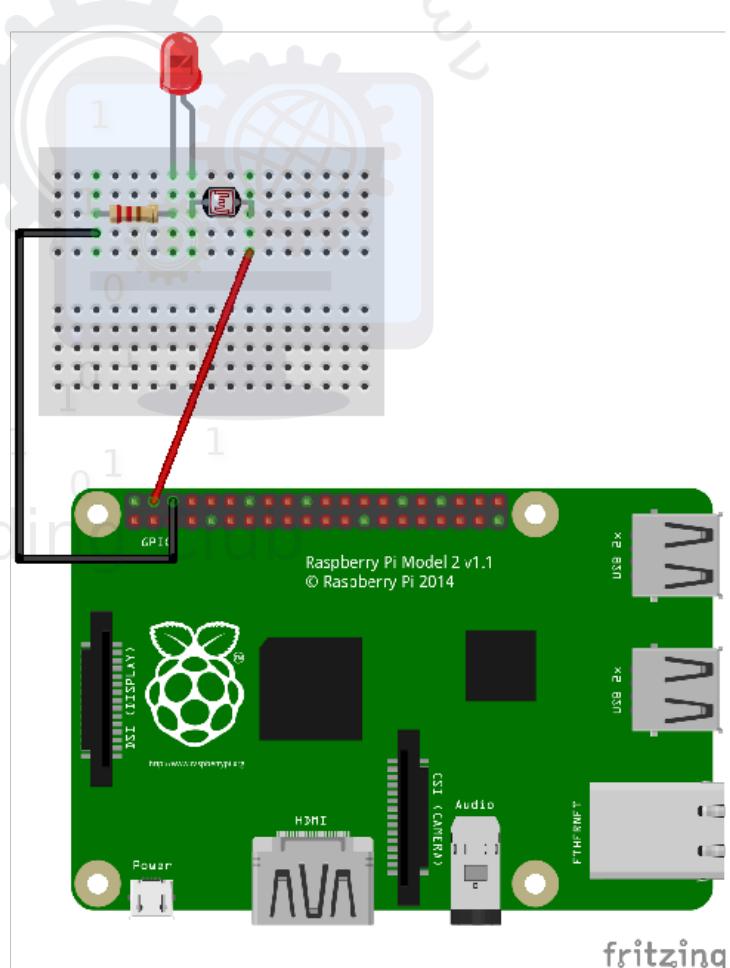
1 Ηχείο

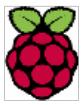
7 Καλώδια (αρσενικό - θηλυκό)

Δραστηριότητα 1 - Σύνδεση

Προσπαθήστε να δημιουργήσετε την διπλανή διάταξη.

Σκεπάστε με το χέρι σας την φωτοαντίσταση και μετά σιγά σιγά ανεβάστε το χέρι σας πιο ψηλά. Τι παρατηρείτε; Μπορείτε να σκεφτείτε σε ποια συσκευή καθημερινής χρήσης έχει εφαρμογή το κύκλωμα αυτό;





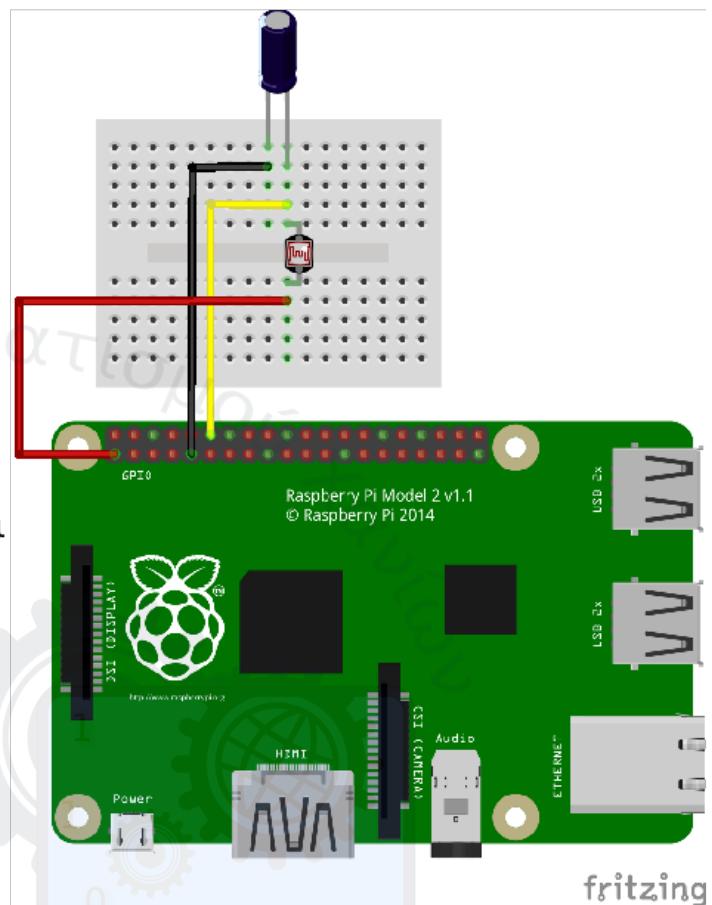
Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Δραστηριότητα 2

Βήμα 1

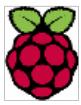
Προσπαθήστε να δημιουργήσετε την διπλανή διάταξη.

Μεγάλη προσοχή όταν πάτε να συνδέσετε τον πυκνωτή. Οι πυκνωτές είναι ευαίσθητοι στην πολικότητα. Πρέπει λοιπόν να συνδέσετε το κοντό ακροδέκτη στη γείωση και τον μακρύ ακροδέκτη στο ρίν που θα περνάει ρεύμα. Στη δική μας περίπτωση το `gpio18`. Αν για κάποιο λόγο το συνδέσετε ανάποδα ο πυκνωτής υπάρχει μεγάλη πιθανότητα να σκάσει, μέρος του οποίου να μπει στα μάτια σας, και να προκληθεί ατύχημα.



fritzing

Chania Coding Club

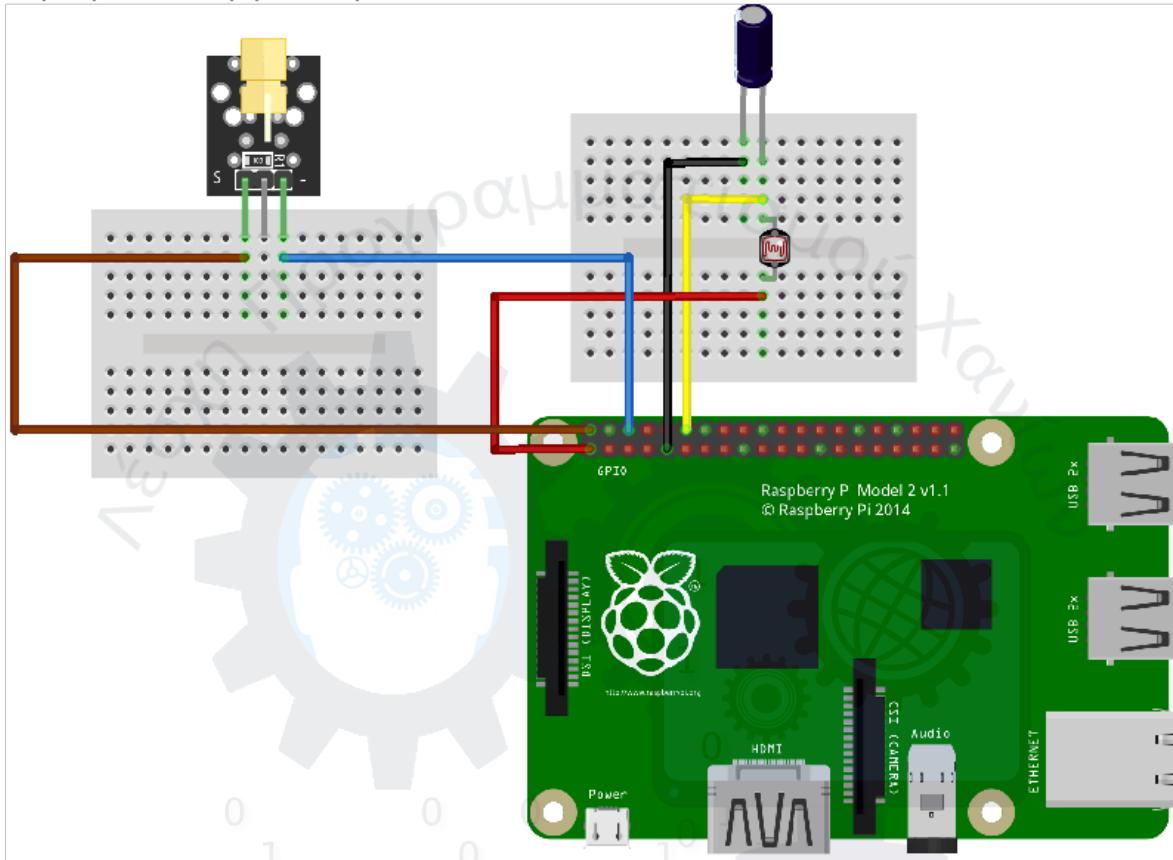


Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Βήμα 2

Σε αυτό το βήμα προσπαθήστε να προσθέσετε και το υπόλοιπο μέρος της συνδεσμολογίας όπου προσθέτουμε τη δέσμη Laser όπως παρακάτω.

Το αρνητικό στη γείωση και το θετικό στο +5 Volt

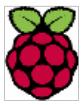


Τι παρατηρείτε μετά από τη σύνδεση;

ChaniaCoding Club



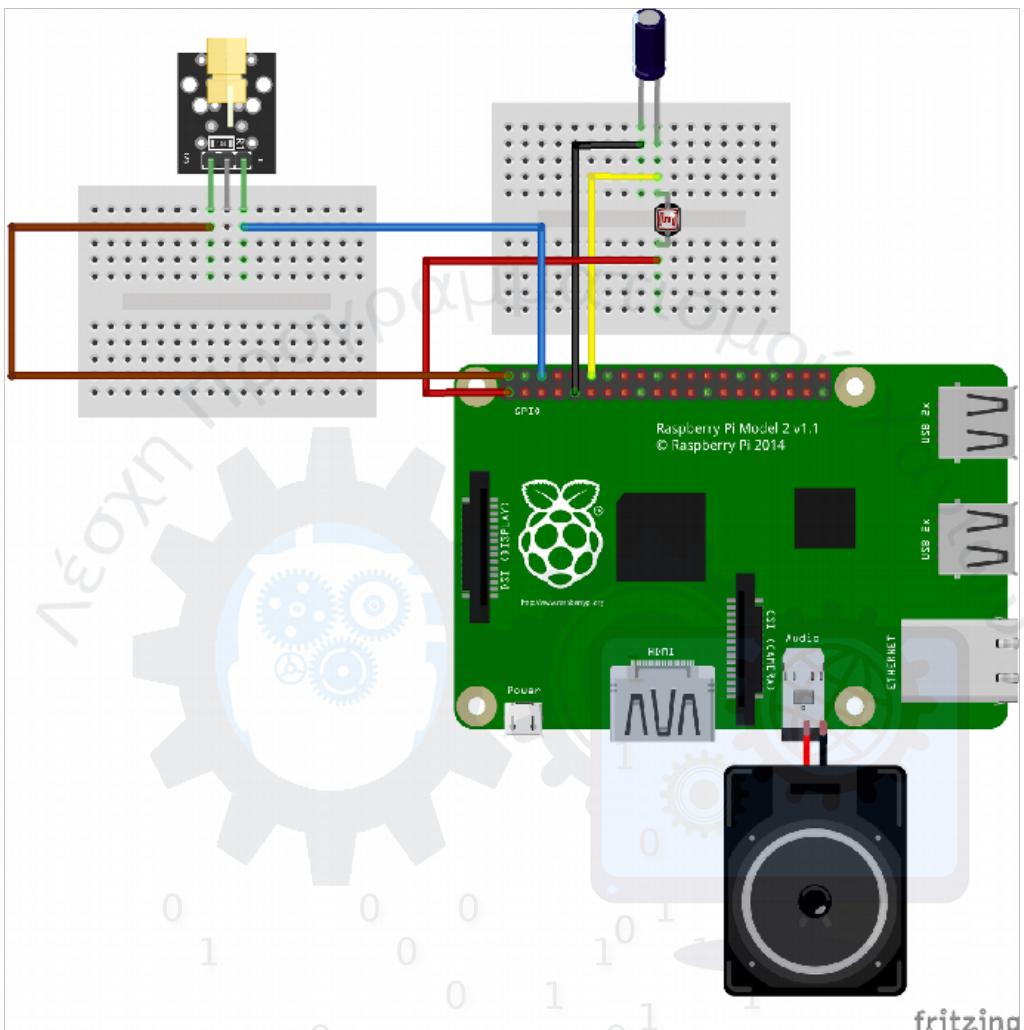
Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



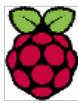
Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Βήμα 3

Σε αυτό το βήμα προσπαθήστε να προσθέσετε και το υπόλοιπο μέρος της συνδεσμολογίας όπου προσθέτουμε το ηχείο όπως παρακάτω.



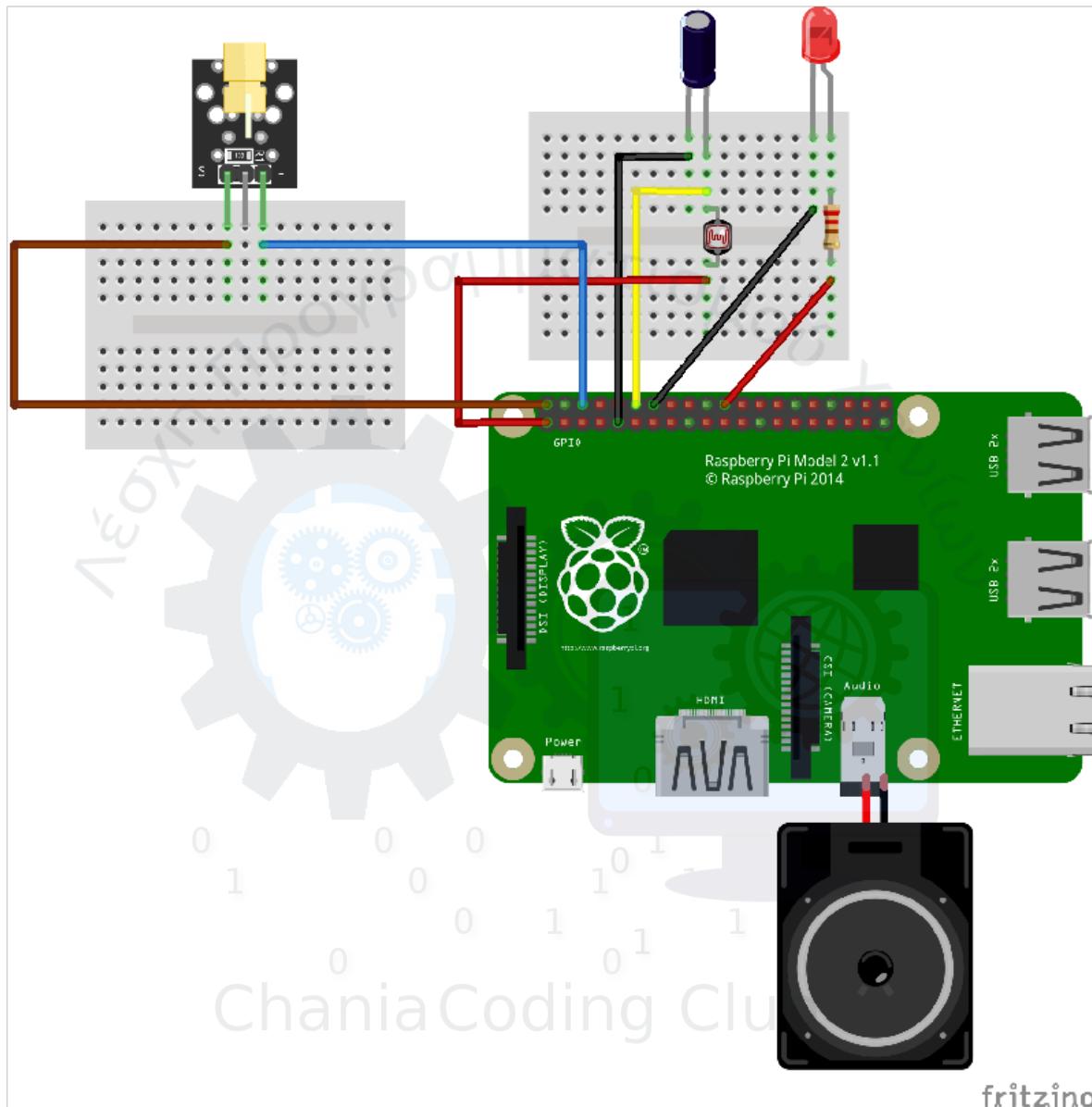
Chania Coding Club

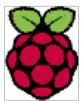


Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Bήμα 4

Σε αυτό το βήμα προσπαθήστε να προσθέσετε και το υπόλοιπο μέρος της συνδεσμολογίας όπου προσθέτουμε ένα led στο gpio25 όπως παρακάτω.





Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Δραστηριότητα 3 - Ωρα για κώδικα

Η φωτοαντίσταση μας δίνει αναλογικές τιμές. Το raspberry pi όμως έχει μόνο ψηφιακές εισόδους με αποτέλεσμα να μην μπορούμε να τη συνδέσουμε έτσι απλά σε κάποια είσοδο και να λειτουργήσει. Θα χρησιμοποιήσουμε μία πλάγια μέθοδο για τη χρήση της φωτοαντίστασης. Θα παρεμβάλουμε λοιπόν ένα πυκνωτή και ουσιαστικά θα μετράμε το χρόνο που χρειάζεται για να φορτίσει και να εκφορτίσει. Όσο πιο γρήγορα φορτίζει τόσο περισσότερο φως έχουμε.

Όταν αρχίσει να περνά ρεύμα σε έναν πυκνωτή, τότε αυτός θα αρχίσει να αποθηκεύει φορτίο. Η τάση στα άκρα του πυκνωτή θα ξεκινήσει να είναι χαμηλή, και θα αυξάνεται καθώς περισσότερο φορτίο συσσωρεύεται.

Με την τοποθέτηση μιας αντίστασης σε σειρά με τον πυκνωτή, μπορείτε να επιβραδύνετε την ταχύτητα με την οποία αποθηκεύει φορτίο. Με μεγάλη αντίσταση, ο πυκνωτής θα αποθηκεύει φορτίο αργά, ενώ μια μικρή αντίσταση θα αποθηκεύει φορτίο γρήγορα.

Πρέπει να μετρήσετε πόσο χρόνο χρειάζεται η τάση του πυκνωτή για να φτάσει πάνω από 1.8V.

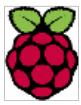
Ευτυχώς, ο περισσότερος από τον πολύπλοκο LDR κώδικα που θα πρέπει να γράψετε για την ανίχνευση των επιπέδων φωτός που λαμβάνονται από το LDR έχει εξαλειφθεί από τη βιβλιοθήκη gpiozero. Η βιβλιοθήκη θα χειριστεί το χρόνο φόρτισης και εκκένωσης του πυκνωτή για εσάς.

Γράψτε τον παρακάτω κώδικα.

Προσθέτουμε το LightSensor από τη βιβλιοθήκη gpiozero. Δημιουργούμε το αντικείμενο ldr που είναι ένα LightSensor αντικείμενο στην GPIO18. Κάνουμε μία ατέρμονη (δίχως τέλος) επανάληψη όπου τυπώνουμε συνεχώς στην οθόνη την τιμή που μας δίνει το ldr.

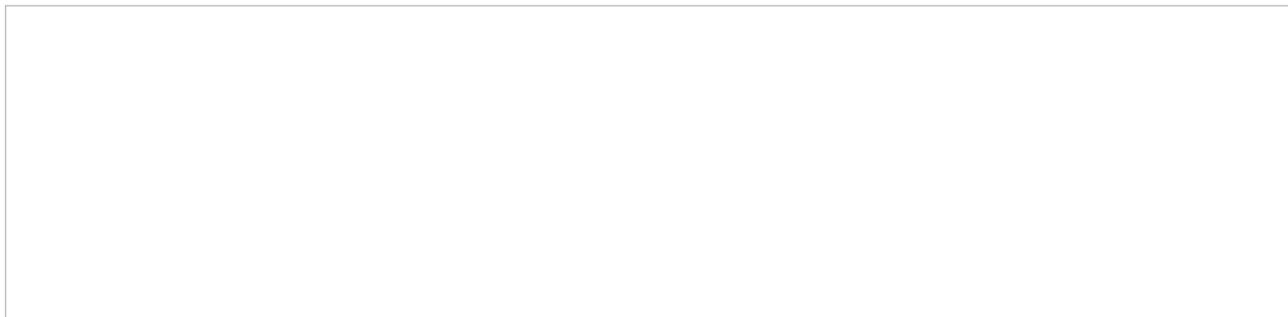
```
from gpiozero import LightSensor  
  
ldr = LightSensor(18)  
while True:  
    print(ldr.value)
```



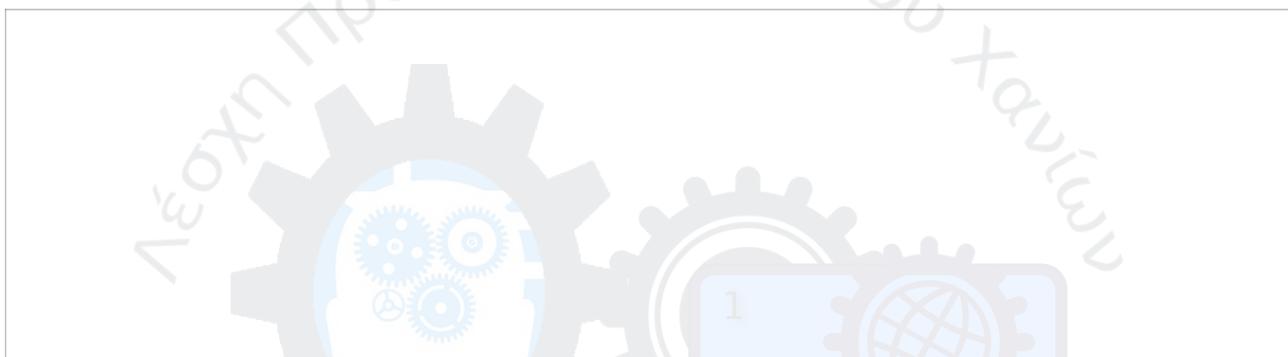


Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Καλύψτε με το χέρι σας την φωτοαντίσταση. Τι παρατηρείτε;

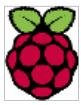


Στρέψτε τη δέσμη laser πάνω στην φωτοαντίσταση. Τι παρατηρείτε;



Τι συμπεράσματα προκύπτουν από τις 2 παραπάνω ενέργειες;





Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Δραστηριότητα 4

Προσθέστε τις απαιτούμενες βιβλιοθήκες και δημιουργήστε κώδικα όπου αν η τιμή του led πέσει κάτω από 0.8 τότε να ανάβει το led για 2 δευτερόλεπτα.

Παρακάτω γράψτε τον κώδικα που χρησιμοποιήσατε.

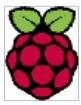
Δραστηριότητα 5

Προσθέστε τις απαιτούμενες βιβλιοθήκες και συμπληρώστε τον κώδικα έτσι ώστε εκτός από το να ανάβει το led για 2 δευτερόλεπτα, να ακούγεται και ένας ήχος σειρήνας. Ο ήχος αυτός είναι αποθηκευμένος στο αρχείο siren.ogg.

Παρακάτω γράψτε τον κώδικα που χρησιμοποιήσατε.



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

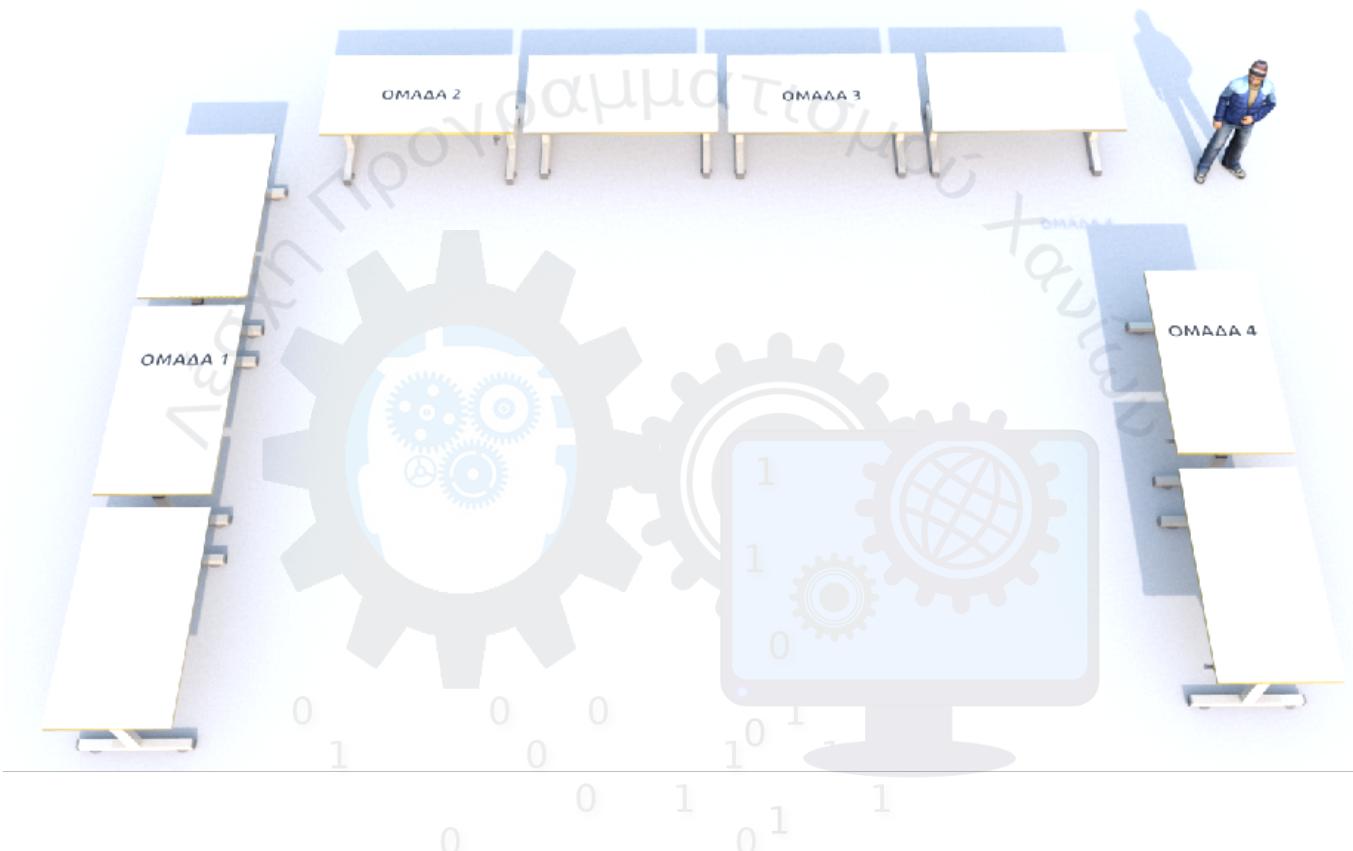
Δραστηριότητα 6

Συνεργαστείτε όλες οι ομάδες και παγιδεύστε την αίθουσα.

Ζωγραφίστε στο παρακάτω πλαίσιο πώς θα φεύγουν οι δέσμες laser από τη μία ομάδα στην άλλη έτσι ώστε να πετύχετε το καλύτερο δυνατό αποτέλεσμα.

Υλοποιήστε το τελικό σχεδιάγραμμα.

Προσοχή στους κανόνες ασφαλείας του laser.



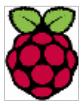
Δραστηριότητα 7

Χρησιμοποιώντας ένα RGBLED προσπαθήστε να κατασκευάσετε κύκλωμα και το κατάλληλο πρόγραμμα σε python όπου θα δημιουργεί ένα χρωματικό ηλιακό ρολόι. Δηλαδή, το χρώμα του RGBLED θα είναι άσπρο όταν το φώς θα είναι έντονο και θα γίνεται όλο και πιο κόκκινο όταν αρχίσει και πέφτει το σκοτάδι.

Παρακάτω γράψτε τον κώδικα που χρησιμοποιήσατε.



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python



Πηγές:

Πρότυπο χρώματος RGB : <https://el.wikipedia.org/wiki/RGB>

GPIO Pin Numbering

<https://www.raspberrypi.org/learning/physical-computing-guide/worksheet/>

Σύνδεση LED στο RaspberryPi:

<https://www.raspberrypi.org/learning/physical-computing-guide/connect-led/>

gpiozero homepage <https://gpiozero.readthedocs.io/en/v1.3.1/>

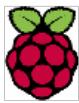
GPIO ZERO: A friendly python API for physical computing:

<https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/>

<https://www.raspberrypi.org/learning/physical-computing-with-python/ldr/>



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Χρήση κουμπιών push-on

Θεωρητικό μέρος

Συναρτήσεις python

Οι συναρτήσεις είναι επαναχρησιμοποιήσιμα μέρη προγραμμάτων. Σας επιτρέπουν να δίνετε ένα όνομα σε ένα σύνολο εντολών και να τρέχετε εκείνο το σύνολο εντολών χρησιμοποιώντας το όνομά τους, οπουδήποτε στο προγραμμά σας και όσες φορές θέλετε. Αυτό είναι γνωστό σαν κλήση (calling) της συνάρτησης.

Οι συναρτήσεις ορίζονται χρησιμοποιώντας τη λέξη κλειδί def, μετά την οποία ακολουθεί ένα όνομα που **ταυτοποιεί** την εκάστοτε συνάρτηση και κατόπιν ακολουθεί ένα ζευγάρι παρενθέσεων που μπορούν να περικλείουν μερικά ονόματα μεταβλητών, και η γραμμή τελειώνει με διπλή τελεία ():.

Παράδειγμα:

```
def sayHello():
    print('Hello World!') # σύνολο εντολών που ανήκουν στη συνάρτηση
# Τέλος της συνάρτησης

sayHello() # κλήση της συνάρτησης
sayHello() # κλήση της συνάρτησης ξανά
```

Έξοδος:

```
Hello World!
Hello World!
```

Πώς δουλεύει:

Ορίζουμε μια συνάρτηση με το όνομα sayHello ακολουθώντας τη σύνταξη όπως εξηγήσαμε παραπάνω. Αυτή η συνάρτηση δεν έχει παραμέτρους γι' αυτό δε δηλώνονται καθόλου μεταβλητές ανάμεσα στις παρενθέσεις. Οι παράμετροι στις συναρτήσεις είναι απλά η είσοδος στη συνάρτηση ώστε να περνάμε διαφορετικές τιμές στη συνάρτηση και να παίρνουμε αντίστοιχα αποτελέσματα.

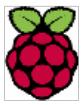
Σημειώστε ότι μπορούμε να καλούμε την ίδια συνάρτηση δύο φορές, δηλαδή δε χρειάζεται να γράφουμε τον ίδιο κώδικα δύο φορές.

Παράμετροι συναρτήσεων

Μια συνάρτηση μπορεί να δεχθεί παραμέτρους, οι οποίες είναι τιμές που δίνετε στη συνάρτηση, έτσι ώστε αυτή να μπορεί να κάνει κάτι αξιοποιώντας αυτές τις τιμές. Αυτές οι παράμετροι μοιάζουν με τις μεταβλητές, διαφέροντας ως



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

προς το ότι οι τιμές αυτών των μεταβλητών ορίζονται όταν καλούμε τη συνάρτηση και τους έχουν ήδη εκχωριθεί τιμές όταν τρέχει η συνάρτηση.

Οι παράμετροι καθορίζονται μέσα στο ζευγάρι των παρενθέσεων στον ορισμό της συνάρτησης και διαχωρίζονται με κόμμα. Όταν καλούμε τη συνάρτηση δίνουμε και τις τιμές με τον ίδιο τρόπο. Σημείωση για την ορολογία που χρησιμοποιείται: οι ονομασίες που δίνετε στον ορισμό της συνάρτησης ονομάζονται παράμετροι ενώ οι τιμές που δίνετε όταν καλείτε τη συνάρτηση ονομάζονται ορίσματα.

Παράδειγμα:

```
def printMax(a, b):
    if a > b:
        print(a, 'είναι το μέγιστο')
    elif a == b:
        print(a, 'είναι ίσο με το', b)
    else:
        print(b, 'είναι το μέγιστο')

printMax(3, 4) # δίνουμε απ' ευθείας κυριολεκτικές τιμές

x = 5
y = 7

printMax(x, y) # δίνουμε μεταβλητές σαν ορίσματα
```

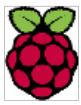
Έξοδος:

4 είναι το μέγιστο
7 είναι το μέγιστο

Πώς δουλεύει:

Εδώ ορίζουμε μια συνάρτηση που ονομάζεται `printMax` με δύο παραμέτρους τις `a` και `b`. Βρίσκουμε το μεγαλύτερο νούμερο χρησιμοποιώντας μια απλή εντολή `if .. else` και μετά τυπώνουμε το μεγαλύτερο νούμερο.

Στην πρώτη χρήση της `printMax`, απευθείας δίνουμε τους αριθμούς, δηλαδή τα ορίσματα. Στη δεύτερη χρήση της, καλούμε τη συνάρτηση χρησιμοποιώντας μεταβλητές. Η `printMax(x, y)` αποδίδει την τιμή του ορίσματος `x` στην παράμετρο `a` και την τιμή του ορίσματος `y` στην παράμετρο `b`. Η συνάρτηση `printMax` λειτουργεί με τον ίδιο τρόπο και στις δύο περιπτώσεις.



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Τοπικές μεταβλητές (Local variables)

Όταν δηλώνετε μεταβλητές μέσα σε ένα ορισμό συνάρτησης, αυτές δεν έχουν καμία σχέση με άλλες μεταβλητές που έχουν την ίδια ονομασία και χρησιμοποιούνται έξω από αυτή τη συνάρτηση, δηλαδή τα ονόματα των μεταβλητών χρησιμοποιούνται μόνο τοπικά στη συνάρτηση. Αυτό ονομάζεται εμβέλεια (scope) των μεταβλητών. Όλες οι μεταβλητές έχουν την εμβέλεια του τμήματος όπου έχουν δηλωθεί, αρχίζοντας από το σημείο στο οποίο ορίζεται το όνομα.

Παράδειγμα:

```
x = 50
```

```
def func(x):
    print('Το x είναι', x)
    x = 2
    print('Άλλαξα το τοπικό x σε', x)

func(x)
print('Το x είναι ακόμα', x)
```

Έξοδος:

```
Το x είναι 50
Άλλαξα το τοπικό x σε 2
Το x είναι ακόμα 50
```

Πώς δουλεύει:

Στη συνάρτηση, την πρώτη φορά που χρησιμοποιούμε την τιμή με το όνομα x, η Python χρησιμοποιεί την τιμή της παραμέτρου που έχει δηλωθεί στη συνάρτηση.

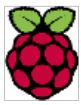
Κατόπιν εκχωρούμε την τιμή 2 στο x. Η ονομασία x είναι τοπική στη συνάρτησή μας. Έτσι όταν αλλάζουμε την τιμή του x στη συνάρτηση, το x που ορίστηκε στο κύριο τμήμα δεν επηρεάζεται.

Στην τελευταία κλήση της συνάρτησης print, παρουσιάζουμε την τιμή x στο κύριο τμήμα και επιβεβαιώνουμε ότι δεν έχει επηρεαστεί.

Χρήση της εντολής global

Εάν θέλετε να εκχωρήσετε μια τιμή σε ένα όνομα που ορίζεται στο κορυφαίο επίπεδο του προγράμματος (δηλαδή όχι μέσα σε κάποιου είδους εμβέλεια όπως σε συναρτήσεις ή κλάσεις), τότε πρέπει να πείτε στην Python ότι το όνομα δεν είναι τοπικό αλλά καθολικό (global). Αυτό γίνεται με τη χρήση της εντολής global. Είναι αδύνατον να εκχωρήσετε μια τιμή σε μια μεταβλητή που ορίζεται





Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

εκτός μιας συνάρτησης χωρίς την εντολή global. Μπορείτε να χρησιμοποιήσετε τις τιμές τέτοιων μεταβλητών που ορίζονται έξω από τη συνάρτηση (υποθέτοντας ότι δεν υπάρχουν μεταβλητές με το ίδιο όνομα μέσα στη συνάρτηση). Ωστόσο, κάτι τέτοιο δεν προτείνεται και πρέπει να αποφεύγεται μιας και δεν είναι ξεκάθαρο για τον αναγνώστη του προγράμματος για το πού βρίσκεται ο ορισμός της μεταβλητής. Χρησιμοποιώντας την εντολή global γίνεται ξεκάθαρο ότι η μεταβλητή βρίσκεται σε ένα εξωτερικό τμήμα εντολών.

Παράδειγμα:

```
x = 50
```

```
def func():
    global x

    print('Το x είναι', x)
    x = 2
    print('Άλλαξα το καθολικό x σε', x)

func()
print('Η τιμή του x είναι', x)
```

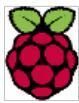
Έξοδος:

```
Το x είναι 50
Άλλαξα το καθολικό x σε 2
Η τιμή του x είναι 2
```

Πώς δουλεύει:

Η εντολή global χρησιμοποιείται για να δηλώσει ότι το x είναι μια καθολική μεταβλητή, γι' αυτό το λόγο όταν εκχωρούμε μια τιμή για το x μέσα στη συνάρτηση, αυτή η αλλαγή απεικονίζεται όταν χρησιμοποιούμε την τιμή του x στο κυρίως τμήμα. Μπορείτε να καθορίσετε περισσότερες από μία καθολικές μεταβλητές χρησιμοποιώντας την ίδια εντολή global. Για παράδειγμα global x, y, z.





Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Πρακτικό μέρος Σύνδεση

Υλικά που θα χρειαστούμε

1 Raspberry pi

1 LED

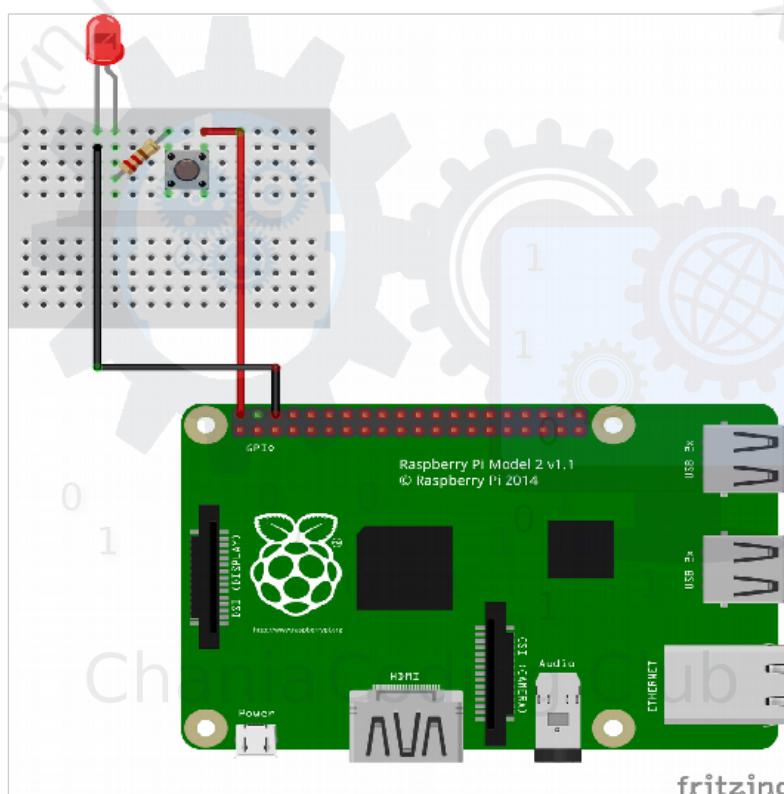
1 αντίσταση 220 ohm

1 Breadboard

7 Καλώδια (αρσενικό - θηλυκό)

Δραστηριότητα 1

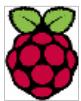
Προσπαθήστε να δημιουργήσετε το παρακάτω κύκλωμα.



Μετά τη σύνδεση πατήστε το κουμπί. Τι παρατηρείτε;



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Δραστηριότητα 2

Τροποποιήστε τη δραστηριότητα 1 συνδέοντας το θετικό του led στο gpio02 και το θετικό του κουμπιού στο gpio18.

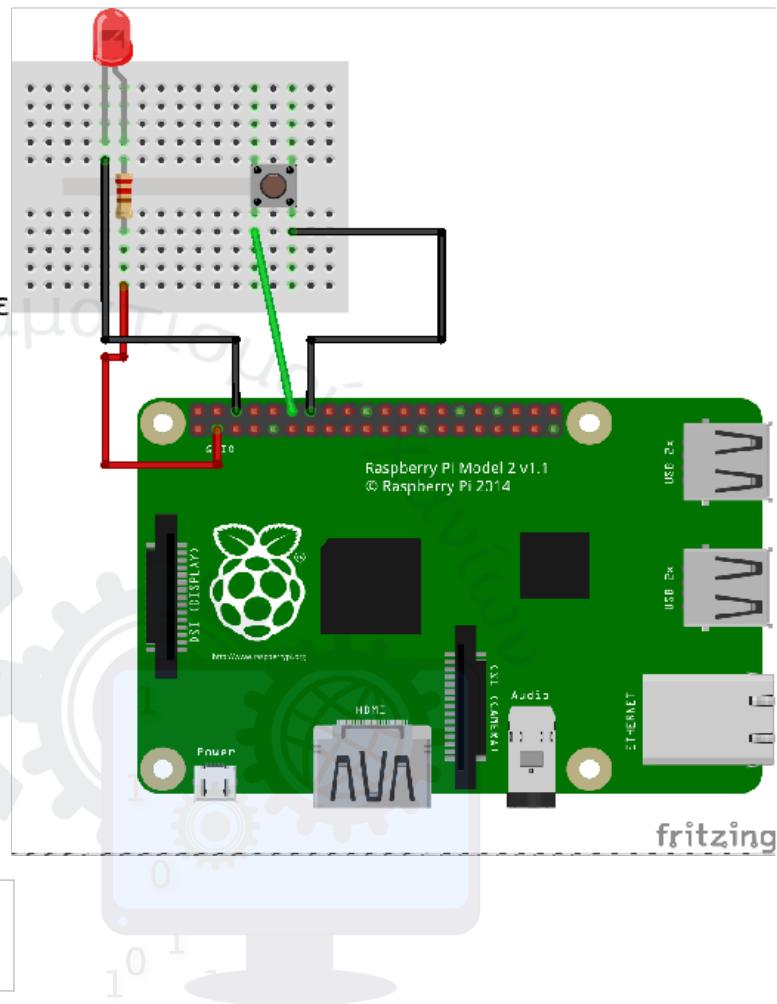
Στην προηγούμενη δραστηριότητα κάναμε το led να ανάβει όσο είναι πιατημένος ο διακόπτης. Σε αυτή τη δραστηριότητα θα μετατρέψουμε με προγραμματισμό τον ίδιο διακόπτη, σε διακόπτη on-off.

Ας γράψουμε λοιπόν λίγο κώδικα.
Πρώτα οι βιβλιοθήκες μας.

Η νέα βιβλιοθήκη στο πρόγραμμα μας είναι η signal και συγκεκριμένα η εντολή pause.

Η pause παγώνει το πρόγραμμά μας μέχρι να δεχθεί κάποιο σήμα. Στην περίπτωσή μας θα είναι το πάτημα του διακόπτη. Δεν λειτουργεί στα windows.

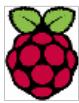
```
from gpiozero import LED, Button
from signal import pause
```



Αρχικοποιούμε τις μεταβλητές μας, και δημιουργούμε ένα αντικείμενο led και ένα αντικείμενο κουμπί (button), στα gpio02 και gpio18 αντίστοιχα. Τη μεταβλητή onoff τη χρησιμοποιούμε σαν δείκτη ετσι ώστε να γνωρίζουμε την κατάσταση του led κάθε φορά. Αν είναι 1 είναι αναμμένο και αν είναι 0 σβηστό.

```
onoff=0
led = LED(02)
onoffbutton = Button(18)
```

Στη συνέχεια δημιουργούμε 2 συναρτήσεις. Την ledon όπου μέσα της έχουμε βάλει εντολές για να ανάψει το led μας, και την ledoff όπου μέσα της έχουμε βάλει εντολές για να σβήσει το led μας.



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

```
def ledon():
```

```
    led.on()
```

```
def ledoff():
```

```
    led.off()
```

Τώρα δημιουργούμε μία συνάρτηση που θα εκτελείτε όταν πατηθεί το κουμπί. Ορίζουμε οτι η onoff μεταβλητή μας είναι global δηλαδή καθολική μεταβλητή και μετά ελέγχουμε την κατάστασή της. Αν είναι ένα εκτελώ τις απαραίτητες ενέργειες για να σβήσω το led, ενώ αν είναι 0 τότε κάνω ό,τι χρειάζεται για να ανάψω το led.

```
def pressed():
```

```
    global onoff
```

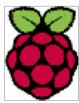
```
    if onoff==1:  
        onoff=0  
        print("off")  
        ledoff()  
    else:  
        onoff=1  
        print ("on")  
        ledon()
```

Οι τελευταίες 2 εντολές χρησιμοποιούνται για τον έλεγχο του κουμπιού. Η πρώτη εκτελεί τη συνάρτηση pressed όταν πατηθεί το κουμπί και η pause παγώνει το πρόγραμμά μας μέχρι να δεχθεί κάποιο σήμα. Στην περίπτωσή μας θα είναι το πάτημα του διακόπτη.

```
onoffbutton.when_pressed = pressed
```

```
pause()
```



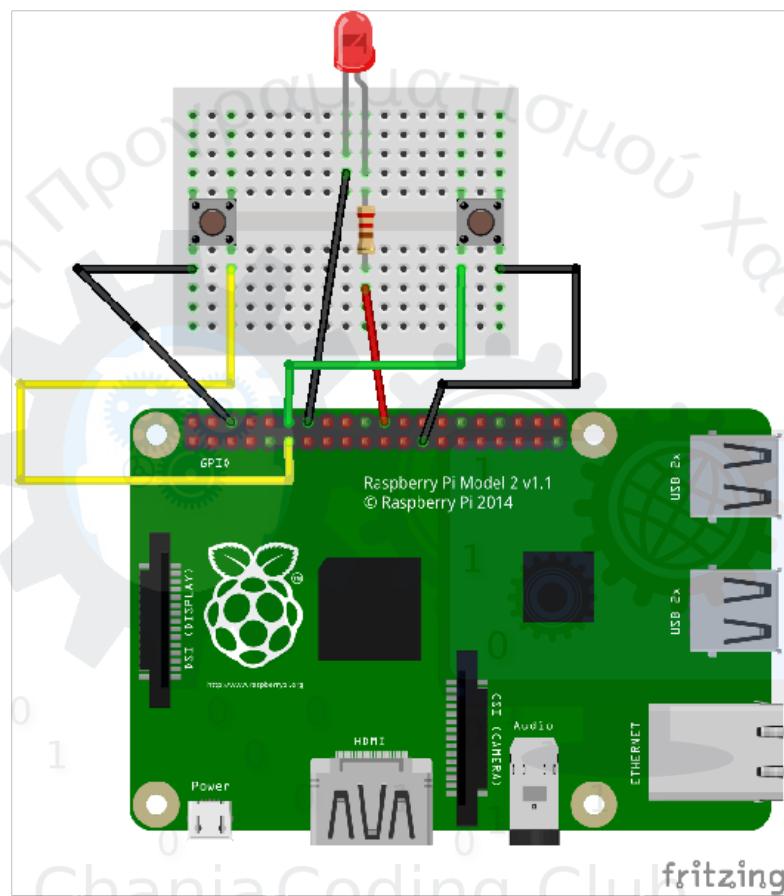


Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Δραστηριότητα 3

Προσπαθήστε να κατασκευάσετε το παρακάτω κύκλωμα, ανά δύο οι ομάδες από κοινού.

Button 1	GPIO 17
Button 2	GPIO 18
LED	GPIO 23

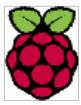


Δραστηριότητα 4 - Παιχνίδι ταχύτητας (reaction game)

Πώς παίζετε

Υπάρχουν 2 κουμπιά, 2 παίκτες και 1 led. Το led ανάβει για τυχαίο χρονικό διάστημα. Οι παίκτες περιμένουν να σβήσει το led. Με το που σβήσει πατούν το κουμπί τους. Ο γρηγορότερος παίκτης κερδίζει το γύρο.





Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Βήμα 1

Πρώτα θα γράψουμε τις βιβλιοθήκες που θα χρησιμοποιήσουμε

```
from gpiozero import LED, Button  
from time import sleep  
from random import uniform  
from signal import pause
```

Βήμα 2

Θα δημιουργήσουμε 2 μεταβλητές για να αποθηκεύουμε τα σκόρ των παικτών.

```
scora=0  
scorb=0
```

Βήμα 3

Δημιουργούμε ένα αντικείμενο led και 2 αντικείμενα κουμπιών στα gpio pins 23, 17, 18 αντίστοιχα.

```
led = LED(23)  
right_button = Button(17)  
left_button = Button(18)
```

Βήμα 4

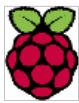
Καλεί τη συνάρτηση ledon για να ανάψει το led και περιμένει αντίδραση από τα 2 κουμπιά. Με το που αντιληφθεί πάτημα καλεί την συνάρτηση pressed.

```
ledon()  
right_button.when_pressed = pressed  
left_button.when_pressed = pressed  
  
pause()
```

Βήμα 5

Σε αυτή τη συνάρτηση ανάβουμε για τυχαίο χρόνο το led. Χρησιμοποιούμε την συνάρτηση uniform(5,10) όπου θα επιστρέψει ένα τυχαίο δεκαδικό αριθμό από 5.1 μέχρι 9.9999999.





Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

```
def ledon():
    led.on()
    sleep(uniform(5, 10))
    led.off()
```

Βήμα 6

Δημιουργούμε τη συνάρτηση pressed και δηλώνουμε μέσα τις μεταβλητές που αποθηκεύουν τα σκορ των παικτών σαν global. Αυτό σημαίνει ότι οποιαδήποτε αλλαγή γίνει στο περιεχόμενο των μεταβλητών μέσα στη συνάρτηση θα επηρεάσει τις μεταβλητές αυτές και έξω από τη συνάρτηση.

```
def pressed(button):
    print(str(button.pin.number) + ' won the game')
    global scora, scrb
    ######
    #####
    sleep(3)
    ledon()
```

Βήμα 7

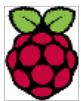
Προσπαθήστε να ολοκληρώσετε το πρόγραμμα και να βάλετε στη θέση που είναι το σύμβολο # τον κώδικα που απαιτείται για να μπορεί να γίνει το παιχνίδι λειτουργικό.

Γράψτε τον κώδικα που χρησιμοποιήσατε παρακάτω

Chania Coding Club



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής



Λέσχη Προγραμματισμού Χανίων - Προγραμματισμός του Raspberry Pi με τη γλώσσα Python

Πηγές:

GPIO Pin Numbering

<https://www.raspberrypi.org/learning/physical-computing-guide/worksheet/>

Σύνδεση LED στο RasberryPi:

<https://www.raspberrypi.org/learning/physical-computing-guide/connect-led/>

gpiozero homepage <https://gpiozero.readthedocs.io/en/v1.3.1/>

GPIO ZERO: A friendly python API for physical computing:

<https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/>

<https://www.raspberrypi.org/learning/physical-computing-with-python/ldr/>

<https://cyberpython.github.io/byte-of-python/functions.html>



Μανούσος Μανούσακας, Χαρά Ξανθάκη, Στέλιος Περράκης, Καθηγητές Πληροφορικής