



# SCRAPY Manual



Co-funded by  
the European Union

Η υποστήριξη της Ευρωπαϊκής Επιτροπής στην παραγωγή της παρούσας έκδοσης δεν συνιστά αποδοχή του περιεχομένου, το οποίο αντικατοπτρίζει αποκλειστικά τις απόψεις των συντακτών, και η Επιτροπή δεν μπορεί να αναλάβει την ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν.

## ΠΡΟΕΙΔΟΠΟΙΗΣΗ

Όταν χρησιμοποιείτε αυτό το προϊόν, λάβετε υπόψη τα ακόλουθα προειδοποιητικά σημεία:

1. Αυτό το προϊόν περιέχει πολλά μικρά μέρη. Η κατάποση ή η ακατάλληλη λειτουργία μπορεί να προκαλέσει σοβαρές λοιμώξεις και θάνατο. Ζητήστε άμεση ιατρική βοήθεια όταν συμβεί το ατύχημα.
2. Απαγορεύστε αυστηρά τη χρήση αυτού του προϊόντος και των εξαρτημάτων του κοντά σε πρίζα AC ή άλλα κυκλώματα σε περίπτωση πιθανού κινδύνου ηλεκτροπληξίας.
3. Απαγορεύστε αυστηρά τη χρήση αυτού του προϊόντος κοντά σε οποιοδήποτε υγρό ή φωτιά.
4. Κρατήστε τα αγωγίμα υλικά μακριά από αυτό το προϊόν.
5. Μην επιτρέπετε σε παιδιά κάτω των 3 ετών να χρησιμοποιούν αυτό το προϊόν χωρίς την επίβλεψη ενηλίκου. Τοποθετήστε αυτό το προϊόν σε θέση όπου δεν μπορούν να φτάσουν τα παιδιά κάτω των 3 ετών.
6. Μην αποθηκεύετε ή χρησιμοποιείτε αυτό το προϊόν σε ακραία περιβάλλοντα όπως υπερβολική ζέστη ή κρύο, υψηλή υγρασία, κάτω από το άμεσο ηλιακό φως κ.λπ.
7. Θυμηθείτε να σπάσετε το κύκλωμα όταν δεν χρειάζεται.
8. Ορισμένα μέρη αυτού του προϊόντος μπορεί να ζεσταθούν όταν χρησιμοποιούνται σε συγκεκριμένα σχέδια κυκλωμάτων, κάτι που είναι φυσιολογικό.
9. Η ακατάλληλη χρήση μπορεί να προκαλέσει υπερθέρμανση.
10. Η χρήση εξαρτημάτων που δεν συμμορφώνονται με τις προδιαγραφές μπορεί να προκαλέσει ζημιά στο προϊόν.



## Εισαγωγή

Το SCRAPY KIT είναι κατασκευασμένο με βάση τη χρήση του μικροελεγκτή Raspberry Pi Pico. Το «SCRAPY KIT» δημιουργείται στο πλαίσιο ενός συγχρηματοδοτούμενου έργου Erasmus+.

Η νέα τάση στην εξ αποστάσεως εκπαίδευση, που έφερε η πανδημία του COVID-19, δημιουργεί ένα κενό στη διδασκαλία και την εξάσκηση δραστηριοτήτων με τη σωματική πληροφορική που φτάνει σε μαθητές που διατρέχουν κίνδυνο χαμηλής απόδοσης σε τέτοια θέματα STEM.

Στόχος του KIT είναι να υποστηρίξει και να προωθήσει την πρακτική εκπαιδευτική μάθηση στους φυσικούς υπολογιστές και στον προγραμματισμό, σε περιπτώσεις εξ αποστάσεως διδασκαλίας και εντός της τάξης. Το KIT παρέχει έναν συνδυασμό αρχών φυσικών υπολογιστών και προγραμματισμού με υλικό και λογισμικό, οδηγώντας σε μια καινοτόμο εμπειρία εκμάθησης.

Το πεδίο εφαρμογής αυτού του Εγχειριδίου είναι να:

- Σας ενημερώσει για τα εξαρτήματα του kit και τη χρήση των κύριων ηλεκτρονικών στοιχείων.
- Σας καθοδηγεί βήμα προς βήμα για να συναρμολογήσετε αποτελεσματικά το KIT, λαμβάνοντας υπόψη τις σχετικές προφυλάξεις.
- Σας παράσχει πειράματα για τη χρήση των εξαρτημάτων και τη σύνδεση με τον μικροελεγκτή Pico.
- Σας παράσχει πειράματα με τις λειτουργίες και το πεδίο εφαρμογής κάθε ηλεκτρονικού στοιχείου.

**Απολαύστε την ανάγνωση και διασκεδάστε μέσω  
πρακτικής εξάσκησης και πειραματισμού με το**

**SCRAPY KIT**

## Πίνακας περιεχομένων

Εισαγωγή.....	1
Πίνακας περιεχομένων.....	2
Περιλαμβάνονται στο SCRAPY KIT .....	4
Επεξήγηση εξαρτημάτων .....	6
1. Τι είναι το breadboard (Πλακέτα Διασύνδεσης); .....	6
2. Τι είναι η αντίσταση;.....	8
3. Τι είναι ένας πυκνωτής;.....	11
4. Τι είναι δίοδος;.....	11
5. Τι είναι ένα καλώδιο jumper (γεφύρωσης/ βραχυκυκλώματος) ;.....	12
Προετοιμασία Πειραμάτων .....	14
Συναρμολόγηση ΚΙΤ.....	22
Βασικά πειράματα.....	26
0. «Hello SCRAPY people!».....	26
1. Ελέγξτε ένα LED.....	28
2. Πιέστε το κουμπί.....	30
3. Buzzer .....	32
4. Ποτενσιόμετρο.....	34
5. RGB LED.....	36
Προχωρημένα πειράματα.....	38
6. LDR Φωτοαντίσταση.....	40
7. Σερβοκινητήρας.....	42
8. Οθόνη OLED I2C SSD1306 .....	44
9. Μονάδα Joystick.....	49

Πειράματα με αισθητήρες .....	51
10. Αισθητήρας σταγόνας βροχής.....	51
11. Αισθητήρας υπερήχων HC-SR04 .....	54
12. Αισθητήρας κίνησης PIR.....	57
13. Αισθητήρας DHT11 .....	59
15. Αισθητήρας φλόγας.....	63
16. Αισθητήρας ανίχνευσης ήχου.....	65
17. Αισθητήρας υγρασίας εδάφους .....	67
18. Αισθητήρας IR υπερύθρων.....	69
ΠΑΡΑΡΤΗΜΑ: Συγκεντρωτικός Πίνακας MicroPython .....	71

## Περιλαμβάνονται στο SCRAPY KIT

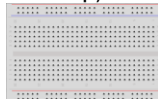
GPIO Breakout Board  
Raspberry Pi Pico (2pcs)



Μονάδα τροφοδοσίας  
MB-102 (1 τεμ.)



Λευκό breadboard 830  
τμχ (1 τμχ) + 400 τμχ (1  
τεμ.)



Κουμπί (1 τεμ.) &  
Καπάκι κουμπιού (1  
τεμ.)



Buzzer (1pc)



Αντιστάσεις 220 Ohm (5  
τμχ) + 1k Ohm (5 τμχ)



Φωτοαντίσταση LDR (1  
τεμ.)



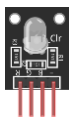
100uF capacitor (1pc)



LED 3 χιλιοστά Μπλε (1  
τμχ), Πράσινο (1 τμχ),  
Κόκκινο (1 τεμ.) Κίτρινο  
(1 τεμ.)



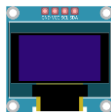
RGB LED 5mm (1pc)



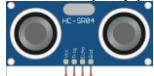
Σερβοκινητήρας SG90 (1  
τεμ.)



OLED I2C ICC (1pc)



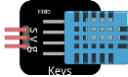
Αισθητήρας υπερήχων  
HC-SR04 (1 τεμ.)

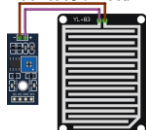
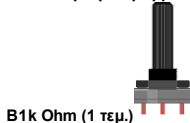


Αισθητήρας ανιχνευτή  
κίνησης PIR HC-SR501  
(1 τεμ.)



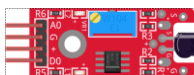
Ψηφιακός αισθητήρας  
θερμοκρασίας και  
υγρασίας DHT11 (1 τεμ.)

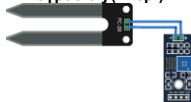


Αισθητήρας σταγόνας  
βροχής (1 τμχ)

 Περιστροφικό  
ποτενσιόμετρο Γραμμικό


B1k Ohm (1 τεμ.)

 Αισθητήρας κραδασμών  
SW-420 (1 τεμ.)

 Αισθητήρας φλόγας (1  
τμχ)

 Αισθητήρας ανίχνευσης  
ήχου υψηλής  
ευαισθησίας (1 τεμ.)

 Υγρόμετρο εδάφους /  
Αισθητήρας ανίχνευσης  
υγρασίας (1 τεμ.)

 Μονάδα αισθητήρα  
υπερύθρων υπέρυθρων  
KY-032 (1 τεμ.)


Μονάδα Joystick (1 τεμ.)


 Καλώδιο USB σε micro-  
USB 1m (1 τεμ.)

 6 x μπαταρίες AA 1,5V (1  
τμχ) + θήκη (1 τεμ.)

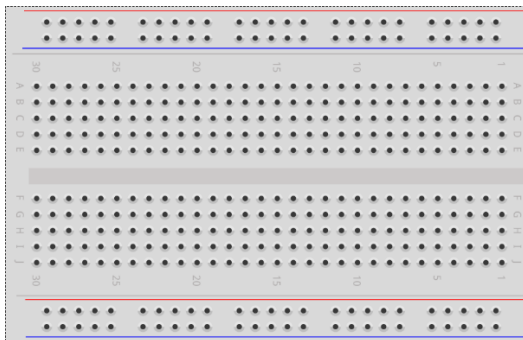

Καλώδια Jumper (6 τεμ.)


 Πλαστικές βίδες (10 τμχ),  
πλαστικά παξιμάδια (6  
τμχ), πλαστικές κολόνες  
(10 τμχ)




## Επεξήγηση εξαρτημάτων

### 1. Τι είναι το breadboard (Πλακέτα Διασύνδεσης);



Το breadboard είναι μια πλαστική πλακέτα με μικροσκοπικές οπές που επιτρέπουν την εύκολη εισαγωγή ηλεκτρονικών εξαρτημάτων (τρανζίστορ, αντιστάσεις, τσιπ, κ.λπ.) για την πρωτότυπη κατασκευή και δοκιμή ενός ηλεκτρονικού κυκλώματος. Το εσωτερικό αποτελείται από σειρές μικροσκοπικών μεταλλικών κλιπ για να συγκρατούν τα καλώδια που πρόκειται να συνδεθούν.

Τα περισσότερα breadboard έχουν σειρές αριθμών, γραμμάτων και σύμβολα συν και πλην γραμμένα πάνω στις σειρές. Ο σκοπός των ετικετών είναι να σας βοηθήσουν να εντοπίσετε συγκεκριμένες οπές στο breadboard, ώστε να μπορείτε να ακολουθείτε τις οδηγίες κατά την κατασκευή ενός κυκλώματος.

Οι μακριές λωρίδες στις 2 πλευρές του breadboard συνήθως σημειώνονται με κόκκινο και μπλε ή κόκκινο και μαύρο και με τα σημάδια συν (+) και πλην (-) αντίστοιχα. Αυτές οι σειρές ονομάζονται ράγες τροφοδοσίας και χρησιμοποιούνται συνήθως για την παροχή ηλεκτρικής ενέργειας στο κύκλωμα όταν συνδέονται σε τροφοδοτικό (μπαταρία ή εξωτερική παροχή).

Η θετική ράγα σημειώνεται με κόκκινο χρώμα, έχει το σύμβολο συν (+) και παρέχει την ισχύ.

Η αρνητική ράγα σημειώνεται με μπλε ή μαύρο, έχει το σύμβολο πλὴν (-) και παρέχει τη γείωση.

Πλεονεκτήματα της χρήσης ενός Breadboard:

- Διευκολύνει τον γρήγορο έλεγχο απλών και πολύπλοκων κυκλωμάτων και την εύκολη επαλήθευση των κυκλωμάτων στο αρχικό τους στάδιο.
- Εύκολη προσαρμογή.
- Ευέλικτο.
- Χωρίς διάνοιξη οπών.
- Δεν απαιτείται συγκόλληση.
- Εύκολος εντοπισμός σφαλμάτων κυκλωμάτων και προγραμμάτων.

## 2. Τι είναι η αντίσταση;



Μια αντίσταση είναι ένα μικρό πακέτο αντίστασης. Η χρήση του σε ένα κύκλωμα μειώνει το ρεύμα κατά ένα ακριβές ποσό. Για να υπολογίσετε την αντοχή μιας αντίστασης υπάρχει ένα σχέδιο από χρωματιστές ταινίες.

Color	1st Band	2nd Band	3rd Band (5-Band Only)	Multiplier (3rd or 4th Band)	Tolerance (Last Band)
Black	0	0	0	1	
Brown	1	1	1	10	± 1%
Red	2	2	2	100	± 2%
Orange	3	3	3	1000	
Yellow	4	4	4	10000	
Green	5	5	5	100000	± 0.5%
Blue	6	6	6	1000000	± 0.25%
Violet	7	7	7	10000000	± 0.1%
Grey	8	8	8		± 0.05%
White	9	9	9		
Gold				0.1	± 5%
Silver				0.01	± 10%
None					± 1%

(Πηγή εικόνας: Future Owns) διαθέσιμη στη διεύθυνση  
<https://www.tomshardware.com/how-to/resistor-color-codes>

### Κοινοί κωδικοί χρωμάτων αντίστασης και οι χρήσεις τους:

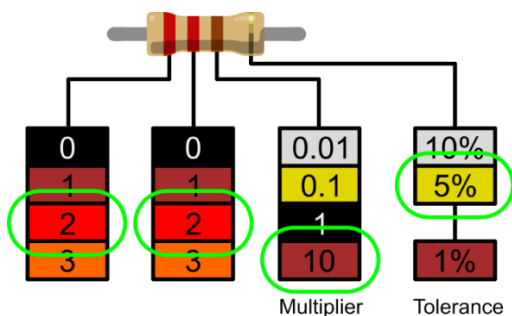
Τύπος αντίστασης	4-Ζώνες Κωδικός χρώματος	5- Ζώνες Κωδικός χρώματος	Κοινή Χρήση
220Ohm	Κόκκινο-Κόκκινο-Καφέ-Χρυσό	Κόκκινο-Κόκκινο-Μαύρο-Μαύρο-Χρυσό	Προστασία από φως LED
1K Ohm (1Kiloohm)	Καφέ-Μαύρο-Κόκκινο-Χρυσό	Καφέ-Μαύρο-Μαύρο-Καφέ-Χρυσό	Προστασία LED, Διαιρέτης τάσης

Οι αντιστάσεις δεν έχουν πολικότητα, επομένως μπορούν να χρησιμοποιηθούν σε οποιονδήποτε προσανατολισμό σε ένα

κύκλωμα. Αλλά για να προσδιορίσουμε τις σωστές τιμές του κωδικού χρώματος της αντίστασης, πρέπει να κατανοήσουμε τις έγχρωμες ζώνες στην αντίσταση.

Σε μια τυπική αντίσταση τεσσάρων ζωνών για δραστηριότητες χόμπι, υπάρχουν τρία ομαδοποιημένα χρώματα. Αυτά είναι τα πρώτα, τα δεύτερα σημαντικά στοιχεία και ο πολλαπλασιαστής. Η τελική ζώνη είναι η αντοχή της αντίστασης, με ένα περιθώριο σφάλματος αν θέλετε. Για την πλειοψηφία των χομπίστων, η αντοχή 5% (Χρυσή ζώνη) είναι πιο κοινή και αποτελεσματική.

### Αντίσταση 220 Ohm (4-Ζώνες)

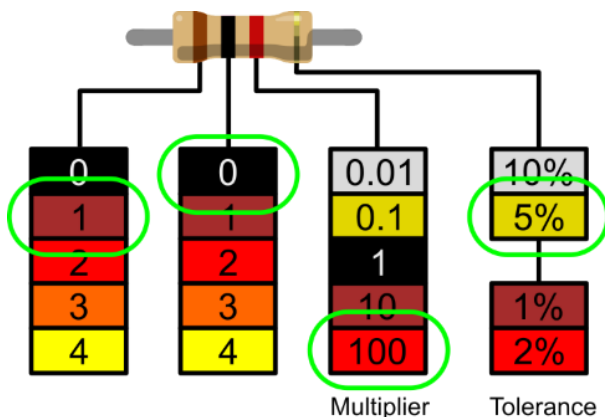


(Πηγή *Eικόνας: Future* διαθέσιμο *στη* διεύθυνση <https://www.tomshardware.com/how-to/resistor-color-codes>)

1. Ο πρώτος σημαντικός αριθμός είναι κόκκινος και χρησιμοποιώντας τον αποκωδικοποιητή μπορούμε να δούμε ότι το κόκκινο έχει τιμή 2.
2. Ο δεύτερος σημαντικός αριθμός είναι επίσης κόκκινος, οπότε μας δίνει 22.
3. Ο πολλαπλασιαστής είναι καφέ και αυτός αποκωδικοποιείται στο 10. Αν πολλαπλασιάσουμε το 22 με το 10, θα έχουμε 220.
4. Η τελική ζώνη, η αντοχή, είναι χρυσή. Η χρυσή είναι 5%, που σημαίνει ότι μπορούμε να δεχτούμε αντίσταση με περιθώριο σφάλματος 5%.

Για τους κατασκευαστές που απαιτούν μεγαλύτερη ακρίβεια, υπάρχουν επίσης αντιστάσεις με πέντε ζώνες που έχουν ένα τρίτο σημαντικό αριθμό. Η πρόσθετη εικόνα παρέχει σαφήνεια που μπορεί να είναι απαραίτητη σε κυκλώματα ευαίσθητα στην αντίσταση, όπως επιστημονικά όργανα και όργανα μηχανικής.

### Αντίσταση 1K Ohm (4-Ζώνες)



(Πηγή *Εικόνα: Tom's Hardware*) διαθέσιμη στη διεύθυνση <https://www.tomshardware.com/how-to/resistor-color-codes>

1. Η 1η γραμμή είναι καφέ και χρησιμοποιώντας τον αποκωδικοποιητή, μπορούμε να δούμε ότι η τιμή είναι 1.
2. Η 2η γραμμή είναι μαύρη, οπότε μας δίνει 10.
3. Ο πολλαπλασιαστής είναι κόκκινος, και αυτός αποκωδικοποιείται στο 100. Αν πολλαπλασιάσουμε το 10 με το 100 θα έχουμε 1000.
4. Η τελική γραμμή, η αντοχή, είναι χρυσή. Το χρυσό είναι 5%, που σημαίνει ότι μπορούμε να δεχτούμε αντίσταση με περιθώριο σφάλματος 5%.

### 3. Τι είναι ένας πυκνωτής;



Ένας πυκνωτής είναι μια συσκευή που αποθηκεύει ηλεκτρική ενέργεια σε ένα ηλεκτρικό πεδίο. Είναι ένα παθητικό ηλεκτρονικό εξάρτημα με δύο τερματικά. Αποτελείται από δύο ηλεκτρικούς αγωγούς που χωρίζονται με απόσταση. Ο χώρος μεταξύ των αγωγών μπορεί να συμπληρωθεί με κενό ή με ένα μονωτικό υλικό γνωστό ως διηλεκτρικό. (Βικιπαίδεια)

Ο πυκνωτής 100 $\mu$ F είναι ένας πυκνωτής ηλεκτρολυτικής αποσύνδεσης. Αυτοί οι πυκνωτές είναι εξαιρετικοί καταστολείς μεταβατικών υπερτάσεων και η χρήση ενός πυκνωτή μεταξύ της ισχύος και της γείωσης του κυκλώματος εξασφαλίζει την ομαλή παροχή ισχύος.

### 4. Τι είναι δίοδος;

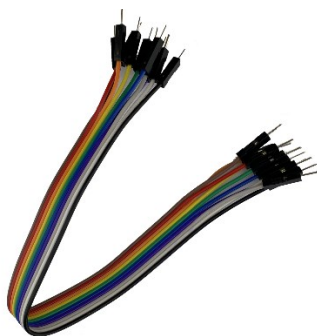


Μια δίοδος είναι ένα ηλεκτρονικό εξάρτημα δύο ακροδεκτών που μεταφέρει ρεύμα κυρίως προς μία κατεύθυνση (ασύμμετρη αγωγιμότητα), έχει χαμηλή (ιδανικά μηδενική) αντίσταση στη μία κατεύθυνση και υψηλή (ιδανικά άπειρη) αντίσταση στην άλλη.

Η πιο κοινή λειτουργία μιας διόδου είναι να επιτρέπει σε ένα ηλεκτρικό ρεύμα να διέρχεται προς μία κατεύθυνση (που ονομάζεται κατεύθυνση προς τα εμπρός της διόδου), ενώ το εμποδίζει προς την αντίθετη κατεύθυνση (την αντίστροφη κατεύθυνση). Ως εκ τούτου, η δίοδος μπορεί να θεωρηθεί ως μια ηλεκτρονική έκδοση μιας βαλβίδας ασφαλείας. Αυτή η συμπεριφορά μονής κατεύθυνσης ονομάζεται αποκατάσταση και

χρησιμοποιείται για τη μετατροπή του εναλλασσόμενου ρεύματος (ac) σε συνεχές ρεύμα (dc). Ως διορθωτικά, οι δίοδοι μπορούν να χρησιμοποιηθούν για εργασίες όπως η εξαγωγή της διαμόρφωσης των ραδιοσημάτων σε ραδιοφωνικούς δέκτες. (Wikipedia <https://en.wikipedia.org/wiki/Diode>)

## 5. Τι είναι ένα καλώδιο jumper (γεφύρωσης/ βραχυκυκλώματος) ;



Τα καλώδια Jumper (γεφύρωσης/ βραχυκυκλώματος) είναι απλά καλώδια που έχουν ακροδέκτες σύνδεσης σε κάθε άκρο, επιτρέποντάς τους να χρησιμοποιούνται για τη σύνδεση δύο σημείων μεταξύ τους χωρίς συγκόλληση. Τα καλώδια αυτά χρησιμοποιούνται συνήθως με breadboards και άλλα εργαλεία δημιουργίας πρωτοτύπων ηλεκτρονικών προκειμένου να είναι εύκολη η αλλαγή ενός κυκλώματος όπως απαιτείται. Η χρωματική διακύμανση των καλωδίων μπορεί να χρησιμοποιηθεί ως πλεονέκτημα για τη διαφοροποίηση μεταξύ των τύπων συνδέσεων, όπως γείωση ή τροφοδοσία.

Τα καλώδια αυτά συνήθως διατίθενται σε τρεις εκδόσεις: αρσενικό σε αρσενικό, αρσενικό σε θηλυκό και θηλυκό σε θηλυκό. Η διαφορά μεταξύ του καθενός είναι στο τελικό σημείο του καλωδίου. Τα αρσενικά άκρα έχουν μια καρφίτσα που προεξέχει και μπορεί να συνδέσει σε αντικείμενα, ενώ τα θηλυκά

άκρα δεν έχουν και χρησιμοποιούνται για να συνδέσουν αντικείμενα. Τα καλώδια από αρσενικό σε αρσενικό είναι τα πιο κοινά. Όταν συνδέετε δύο θύρες σε ένα breadboard, απαιτείται κυρίως καλώδιο αρσενικό σε αρσενικό. (<https://blog.sparkfuneducation.com/what-is-jumper-wire>)



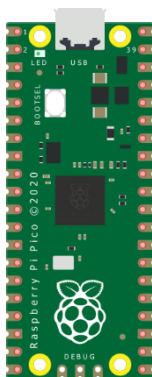
## Προετοιμασία Πειραμάτων

### 1. Διαβάστε πριν ξεκινήσετε

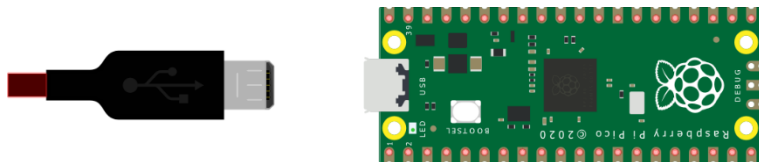
**ΣΗΜΕΙΩΣΗ:** Δεδομένου ότι τα εμπλεκόμενα πειράματα είναι όλα πειράματα κυκλωμάτων, η λανθασμένη σύνδεση ή ένα βραχυκύκλωμα μπορεί να προκαλέσει βλάβη στην πλακέτα RPi Pico. Παρακαλούμε, ελέγχετε πάντα ξανά το κύκλωμα πριν συνδέσετε το τροφοδοτικό.

### 2. Ο Μικροελεγκτής Raspberry Pi Pico

Αυτό είναι ένα Raspberry Pi Pico:

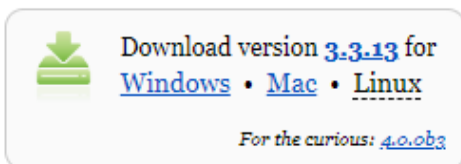


Συνδέστε το καλώδιο micro-USB στη θύρα στην αριστερή πλευρά της πλακέτας.



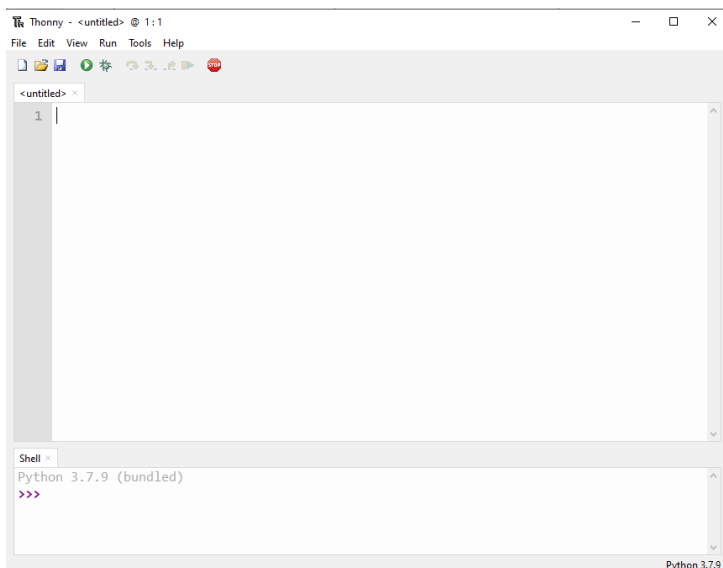
### 3. Εγκαταστήστε το Thonny IDE

Επισκεφτείτε τη διεύθυνση <https://thonny.org> και επιλέξτε το κατάλληλο λειτουργικό σύστημα. Ακολουθήστε τις οδηγίες για να ολοκληρώσετε την εγκατάσταση.



Σε αυτό το εγχειρίδιο, όλα τα πειράματα προγραμματίζονται σε Windows 10, χρησιμοποιώντας ένα Raspberry Pi Pico και το κατάλληλο υλικολογισμικό (firmware).

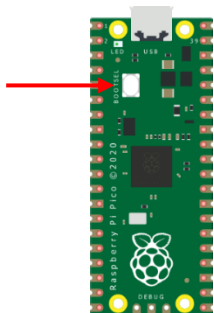
Αφού ολοκληρωθεί η εγκατάσταση, ανοίξτε το Thonny από τον υπολογιστή σας.



#### 4. Εγκατάσταση υλικολογισμικού (firmware)

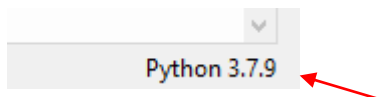
Το Raspberry Pi Pico μπορεί να προγραμματιστεί χρησιμοποιώντας μια παραλλαγή της Python, που ονομάζεται MicroPython. Για να χρησιμοποιήσετε τη MicroPython στο Pico, πρέπει πρώτα να εγκαταστήσετε το λογισμικό της.

Βήμα 1: Βρείτε το κουμπί BOOTSEL στο Raspberry Pi Pico σας.

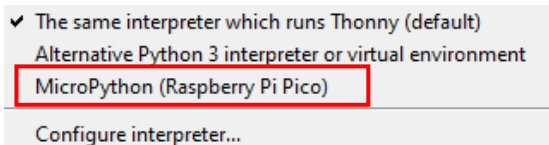


Βήμα 2: Πατήστε το κουμπί BOOTSEL και κρατήστε το ενώ συνδέετε το άλλο άκρο του καλωδίου micro-USB στον υπολογιστή σας.

Βήμα 3: Στην κάτω δεξιά γωνία του Thonny θα δείτε την έκδοση της Python που χρησιμοποιείτε αυτήν τη στιγμή.

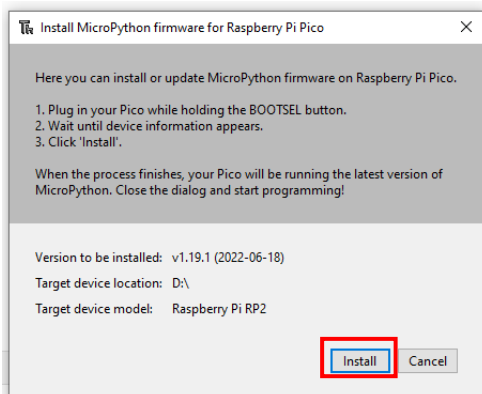


Κάντε κλικ στην έκδοση Python και επιλέξτε το MicroPython (Raspberry Pi Pico)

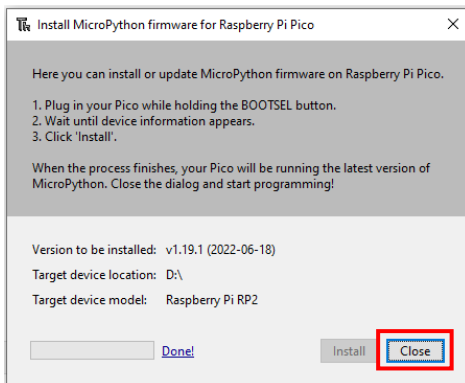


Εάν δεν βλέπετε αυτήν την επιλογή, βεβαιωθείτε ότι το καλώδιο είναι σωστά συνδεδεμένο στο Pico ή/και στον υπολογιστή σας.

Βήμα 4: Θα εμφανιστεί ένα παράθυρο διαλόγου, το οποίο σας ζητά να εγκαταστήσετε την πιο πρόσφατη έκδοση υλικολογισμικού στο Pico σας. Κάντε **κλικ** στο κουμπί Εγκατάσταση για να ανιγράψετε το υλικολογισμικό στο Pico σας.



Βήμα 5: Περιμένετε να ολοκληρωθεί η εγκατάσταση και κάντε κλικ στο **Κλείσιμο**.



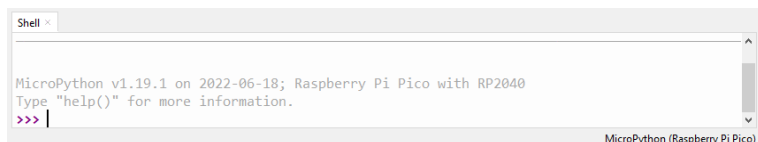
Δεν χρειάζεται να επαναλαμβάνετε τη διαδικασία κάθε φορά που συνδέετε το Raspberry Pi Pico στον υπολογιστή σας, οπότε την επόμενη φορά, απλώς συνδέστε το και είστε έτοιμοι.

## 5. Εισαγωγή στον Προγραμματισμό με MicroPython

Τώρα θα χρησιμοποιήσετε το Thonny IDE για να εκτελέσετε έναν απλό κώδικα Python για να εξοικειωθείτε με το Thonny Shell και το MicroPython.

Πρώτα βεβαιωθείτε ότι το Raspberry Pi Pico είναι συνδεδεμένο στον υπολογιστή σας και ότι έχετε επιλέξει τον διεργαστή MicroPython όπως υποδεικνύεται στην προηγούμενη ενότητα.

Ο πίνακας Shell στο κάτω μέρος του επεξεργαστή Thonny θα πρέπει να μοιάζει με αυτό:



```
Shell <
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> |
```

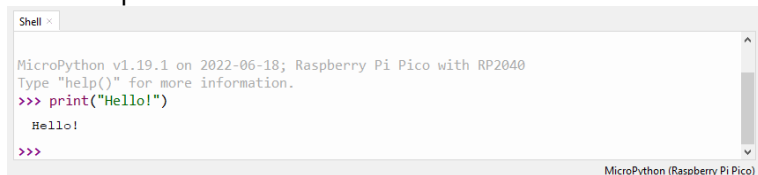
MicroPython (Raspberry Pi Pico)

Ο Thonny είναι έτοιμος να επικοινωνήσει με το Pico σας χρησιμοποιώντας το πλαίσιο REPL (read-eval-print loop), το οποίο σας επιτρέπει να γράψετε κώδικα απευθείας στο Shell και να λάβετε έξοδο.

Πληκτρολογήστε την ακόλουθη εντολή:

```
print("Hello!")
```

Στη συνέχεια, πατήστε το πλήκτρο Enter και δείτε το ακόλουθο αποτέλεσμα:



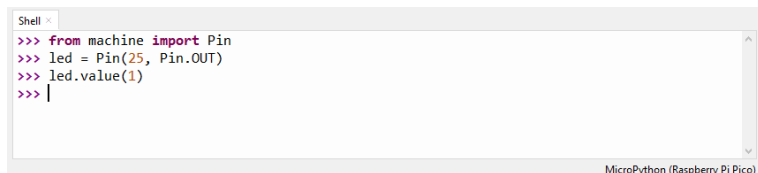
```
Shell <
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> print("Hello!")
Hello!
>>>
```

MicroPython (Raspberry Pi Pico)

Η MicroPython σας επιτρέπει να προσθέσετε ενότητες ειδικά για το υλικό, όπως `machine`, που μπορείτε να χρησιμοποιήσετε για να προγραμματίσετε το Pico σας. Στο παρακάτω παράδειγμα, θα χρησιμοποιήσετε τη μονάδα `machine` για να ενεργοποιήσετε το ενσωματωμένο LED του Pico.

Γράψτε τον παρακάτω κώδικα στο Thonny Shell:

```
from machine import Pin
led = Pin(25, Pin.OUT)
led.value(1)
```



```
Shell
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>> |
```

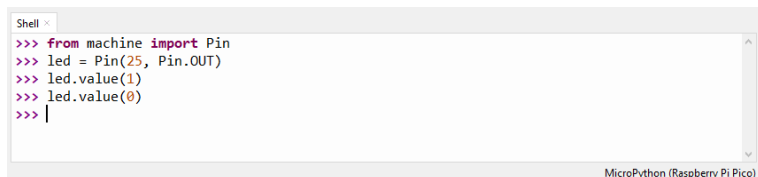
MicroPython (Raspberry Pi Pico)

Πατήστε το πλήκτρο Enter και αμέσως θα ανάψει το ενσωματωμένο LED του Pico.



Για να απενεργοποιήσετε το LED γράψτε τον ακόλουθο κώδικα:

```
led.value(0)
```



```
Shell
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>> led.value(0)
>>> |
```

MicroPython (Raspberry Pi Pico)

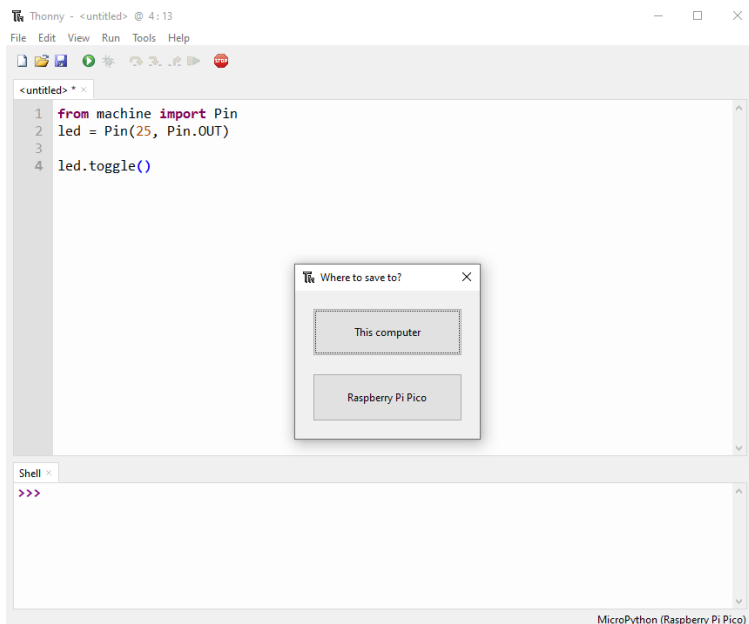
Για το υπόλοιπο αυτής της ενότητας, θα γράψετε το πρώτο σας «πραγματικό» πρόγραμμα που θα κάνει το ενσωματωμένο LED να αναβοσβήνει κάθε φορά που εκτελείτε το πρόγραμμά σας. Το Thonny Shell είναι χρήσιμο για να δοκιμάσετε γρήγορες εντολές και να βεβαιωθείτε ότι όλα λειτουργούν σωστά. Ωστόσο, τα μεγαλύτερα προγράμματα θα πρέπει να αποθηκεύονται σε ένα αρχείο .py. Χρησιμοποιώντας το Thonny, μπορείτε να

αποθηκεύστε προγράμματα απευθείας στο Raspberry Pi Pico και στη συνέχεια να τα εκτελέσετε.

Ανοίξτε το Thonny Python και στο κύριο παράθυρο του επεξεργαστή γράψτε τον ακόλουθο κώδικα:

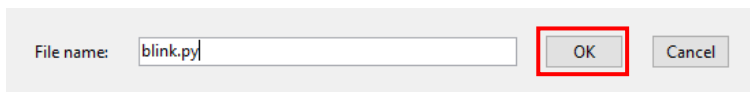
```
from machine import Pin
led = Pin(25, Pin.OUT)
led.toggle()
```

Τώρα, αποθηκεύστε το πρόγραμμά σας κάνοντας κλικ στο εικονίδιο Αποθήκευση στην επάνω αριστερή πλευρά ή πατώντας Ctrl+S στο πληκτρολόγιό σας.



Το Thonny θα σας ρωτήσει πού θέλετε να αποθηκευτεί το πρόγραμμά σας. Επιλέξτε το **Raspberry Pi Pico**. Αποθηκεύστε το αρχείο ως *blink.py* και κάντε κλικ στο **OK**. Πρέπει πάντα να

προσθέτετε την επέκταση .py έτσι ώστε να αναγνωρίζει το αρχείο ως αρχείο Python.



Τώρα, κάθε φορά που κάνετε κλικ στο εικονίδιο Αναπαραγωγή, θα πρέπει να βλέπετε το ενσωματωμένο LED να ανάβει και να σβήνει.

Πηγαίνοντας τον κώδικά σας ένα βήμα παραπέρα, μπορείτε να κάνετε το ενσωματωμένο LED να αναβοσβήνει με συγκεκριμένο ρυθμό.

Γράψτε τον παρακάτω κώδικα και αποθηκεύστε το πρόγραμμά σας χρησιμοποιώντας το ίδιο όνομα όπως παραπάνω.

```
from machine import Pin
from time import sleep
led = Pin(25, Pin.OUT)
while True:
    led.toggle()
    sleep(1)
```

Τώρα όταν εκτελείτε το πρόγραμμα, το ενσωματωμένο LED θα αναβοσβήνει κάθε 1 δευτερόλεπτο μέχρι να σταματήσουμε το πρόγραμμα. Για να σταματήσετε το πρόγραμμα μπορείτε να κάνετε κλικ στο εικονίδιο STOP ή να πατήσετε Ctrl+C στο πληκτρολόγιό σας.

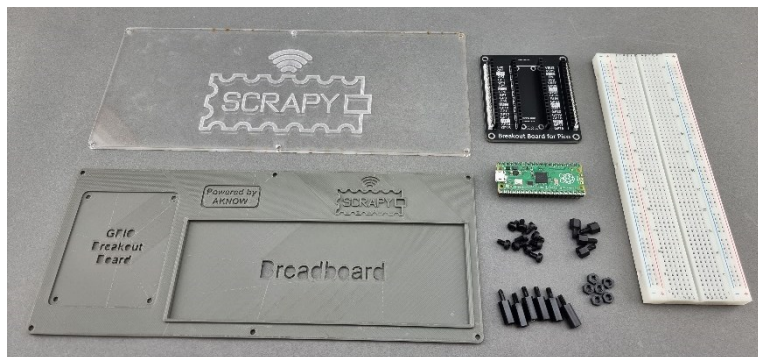
Σε μελλοντικά πειράματα, θα μάθουμε πώς να προσθέτουμε και να ελέγχουμε άλλα ηλεκτρονικά και αισθητήρες και να δημιουργούμε προγράμματα που μπορούν να επικοινωνούν μεταξύ τους.



## Συναρμολόγηση κιτ

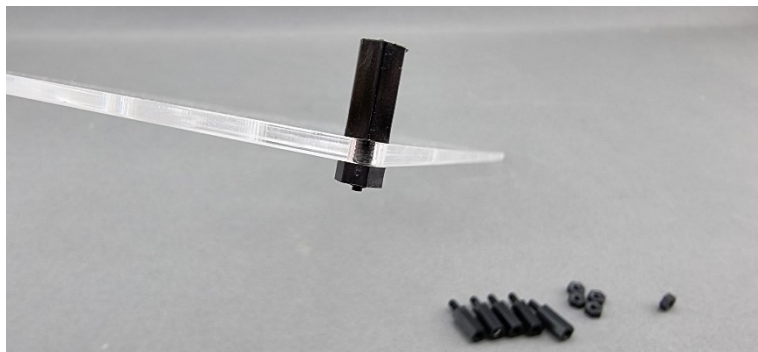
Το κιτ SCRAPY περιλαμβάνει τα ακόλουθα εξαρτήματα:

- 1x 3D εκτυπωμένη βάση
- 1 x τεμάχιο πλεξιγκλάς
- 1 x Raspberry Pi Pico
- 1 x πλακέτα διακοπής GPIO
- 1 x Breadboard (830 τμχ)
- 10 x Πλαστικές βίδες
- 6 x 12mm πλαστικές κολόνες
- 4 x 6mm πλαστικές κολόνες
- 6 x Πλαστικά nuts

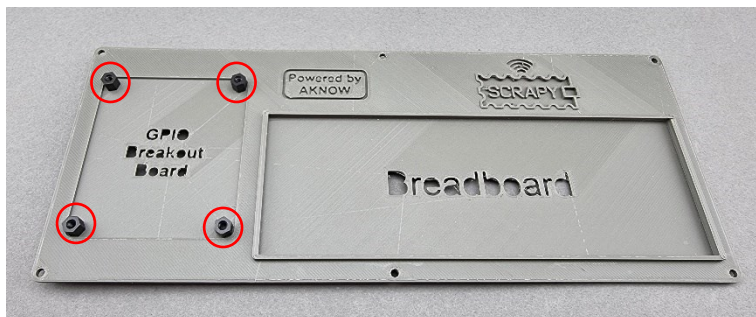


Η διαδικασία συναρμολόγησης είναι απλή και μπορεί να ολοκληρωθεί σε 6 βήματα:

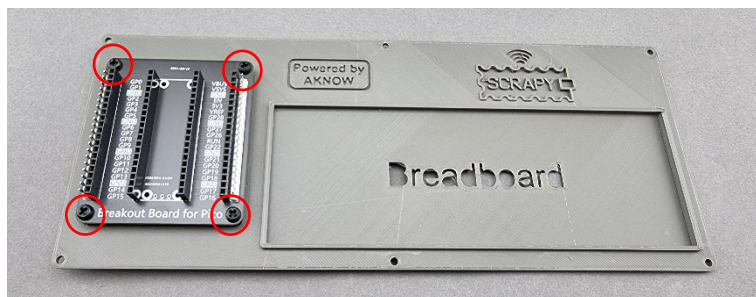
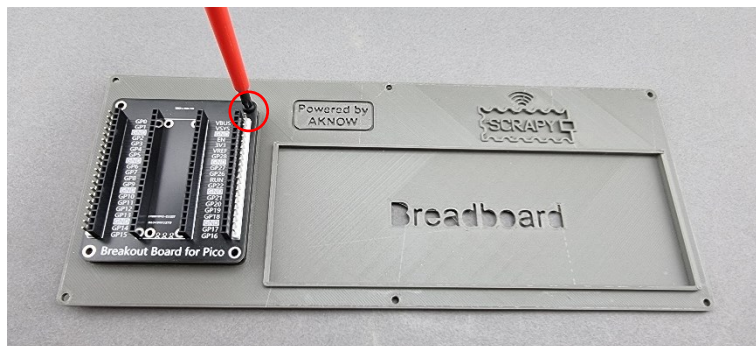
Βήμα 1: Τοποθετήστε τις πλαστικές κολόνες 6 x 12 mm με τα πλαστικά παξιμάδια 6 x στο κομμάτι plexiglass.



Βήμα 2: Στο τρισδιάστατο εκτυπωμένο κομμάτι, τοποθετήστε 4 x 6 mm πλαστικές κολόνες στην ενότητα "GPIO Breakout Board".

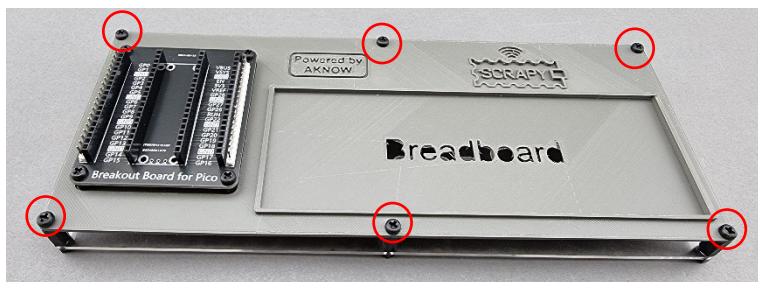


Βήμα 3: Τοποθετήστε την πλακέτα GPIO Breakout στις 4 κολόνες χρησιμοποιώντας 4 πλαστικές βίδες.

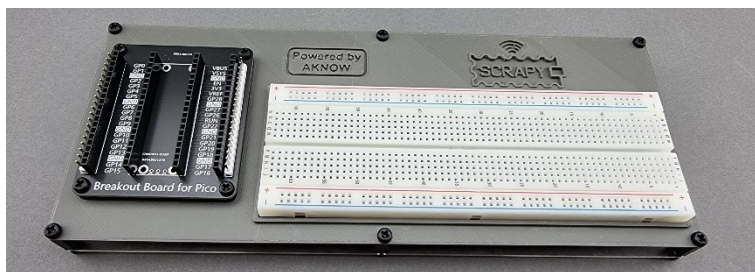


Βήμα 4: Τοποθετήστε το 3D εκτυπωμένο αντικείμενο με το αντικείμενο plexiglass χρησιμοποιώντας 6 πλαστικές βίδες στις 6 πλαστικές κολόνες που τοποθετήσατε στο Βήμα 1.

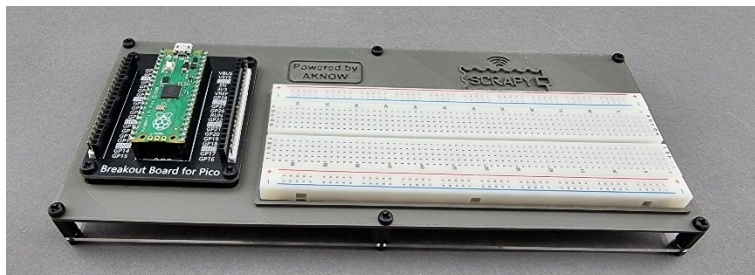




Βήμα 5: Αφαιρέστε την προστατευτική ταινία κάτω από το breadboard των 830 τμχ και τοποθετήστε την στο τμήμα "Breadboard" του κιτ. Βεβαιωθείτε ότι η θετική (+) πλευρά του breadboard είναι στην κορυφή, όπως δείχνει η παρακάτω εικόνα.



Βήμα 6: Τοποθετήστε το Raspberry Pi Pico πάνω από το GPIO Breakout Board και πιάστε το επάνω του μέχρι να εισαχθούν σωστά όλες οι ακίδες GPIO. Βεβαιωθείτε ότι η υποδοχή micro-USB βρίσκεται στην επάνω πλευρά, όπως δείχνει η εικόνα.



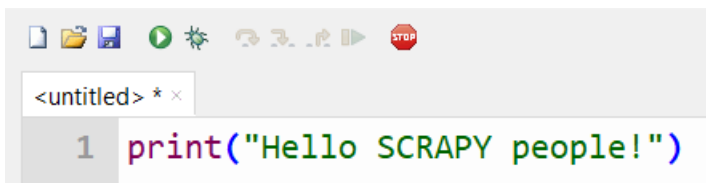
## Βασικά πειράματα

### 0. «Hello SCRAPY people!»

Σε αυτό το βασικό πείραμα, θα μάθουμε πώς να εμφανίζουμε ένα απλό μήνυμα στο Thonny χρησιμοποιώντας την γλώσσα προγραμματισμού Python.

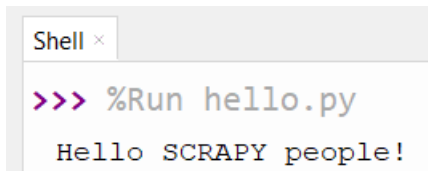
#### Η λειτουργία print()

1. Μεταβείτε στον επεξεργαστή Thonny, γράψτε τον παρακάτω κώδικα και πατήστε το εικονίδιο αναπαραγωγής. Το πρόγραμμα Thonny θα σας ζητήσει να αποθηκεύσετε πρώτα το πρόγραμμά σας. Αποθηκεύστε το με το όνομα `hello.py`.



```
<untitled> * x
1 print("Hello SCRAPY people!")
```

2. Ελέγξτε το παράθυρο shell.



```
Shell x
>>> %Run hello.py
Hello SCRAPY people!
```

3. Καλή δουλειά! Μόλις δημιουργήσατε το πρώτο σας πρόγραμμα Python.

Η λειτουργία `print()` είναι μια ενσωματωμένη συνάρτηση Python που μας επιτρέπει να εκτυπώνουμε κείμενο στο shell. Μπορεί επίσης να πάρει παραμέτρους. Δημιουργήστε ένα νέο πρόγραμμα και αντιγράψτε τον παρακάτω κώδικα, μετά πατήστε play και δείτε πώς εμφανίζεται το κείμενο στο shell.

```

hello.py x
1 print(1,2,3,4,5) #This is a comment!
2 print("I am ",2,"awesome") #1 line
3 print("Python is") #1 line
4 print("amazing") #1 line
5 print("I cant wait.....\n to learn more") # 2 lines of output!
    
```

Όπως μπορείτε να δείτε από το παράθυρο shell, κάθε λειτουργία `print()` εμφανίζει το κείμενο σε ξεχωριστή γραμμή. Ωστόσο, εάν χρησιμοποιείτε το `"\n"` (χαρακτήρας νέας γραμμής), μπορείτε να αλλάξετε γραμμή στην ίδια πρόταση .

```

Shell x
>>> %Run hello.py

1 2 3 4 5
I am 2 awesome
Python is
amazing
I cant wait.....
  to learn more
    
```

## Άσκηση

Χρησιμοποιήστε τη λειτουργία `print()` για να εμφανίσετε σε 3 ξεχωριστές γραμμές "Goodbye SCRAPY people!" χρησιμοποιώντας μόνο μία δήλωση εκτύπωσης.

## 1. Ελέγξτε ένα LED

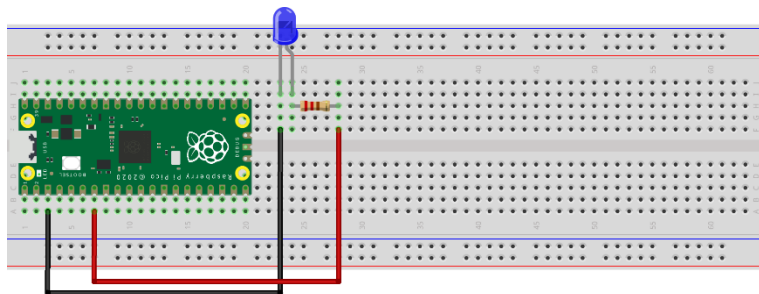
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε ένα φως LED. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως) επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `led.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 2 x καλώδιο jumper αρσενικό σε αρσενικό
- 1 x breadboard πλήρους μεγέθους
- 1 x LED (οποιοδήποτε χρώμα)
- 1 x αντίσταση 220 Ohm

### Διάγραμμα συνδεσμολογίας



fritzing

- συνδέστε το μεγαλύτερο άκρο (+) της λυχνίας LED σε αντίσταση 220 Ohm
- συνδέστε την αντίσταση στο GPIO5 (κόκκινο καλώδιο)
- συνδέστε το μικρότερο άκρο (-) του LED σε μια ακίδα GND

## Κώδικας

Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico :: /led.py @ 9:1
File Edit View Run Tools Help

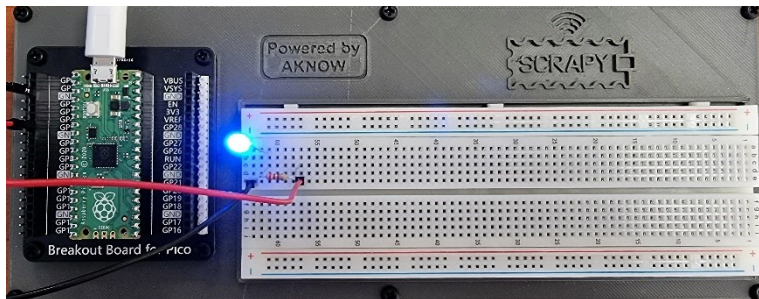
[ led.py ] x
1 from machine import Pin #import Pin from library machine
2 from time import sleep #import sleep from library time
3
4 led = Pin(5, Pin.OUT)
5
6 while True:
7     led.toggle()
8     sleep(1)
9 |

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
    
```

MicroPython (Raspberry Pi Pico)

## Παράδειγμα

Εικόνα της εμφάνισης του κυκλώματος χρησιμοποιώντας το παρεχόμενο υλικό:





## 2. Πιέστε το κουμπί

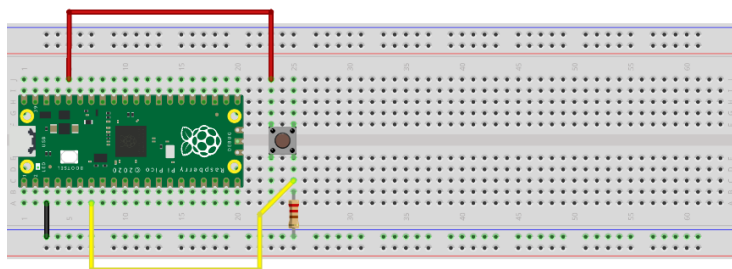
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε ένα κουμπί. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας κάτω από το όνομα `button.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κουμπί
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x Αντίσταση 220 Ohm
- 1 x Καπάκι κουμπιού

### Διάγραμμα συνδεσμολογίας

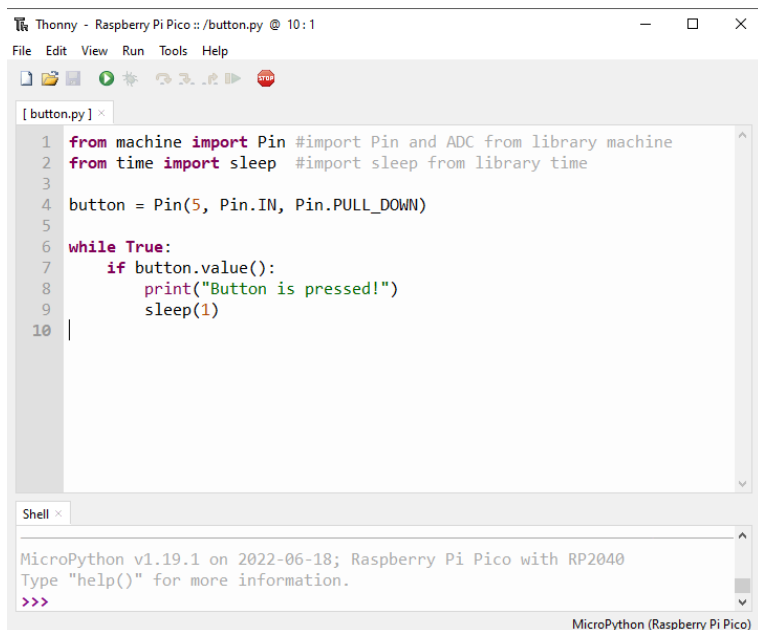


fritzing

- συνδέστε την πάνω αριστερή πλευρά του κουμπιού στην ακίδα 3v3 (κόκκινο καλώδιο)
- συνδέστε την κάτω δεξιά πλευρά του κουμπιού στο GPIO5 (κίτρινο καλώδιο)
- συνδέστε έναν πείρο GND στη γραμμή (-) (μαύρο καλώδιο)
- συνδέστε την αντίσταση 220 Ohm στη γραμμή (-) και στην κάτω δεξιά πλευρά του κουμπιού

## Κώδικας

Κώδικας MicroPython για το πείραμα:



```

Thonny - Raspberry Pi Pico ::/button.py @ 10:1
File Edit View Run Tools Help

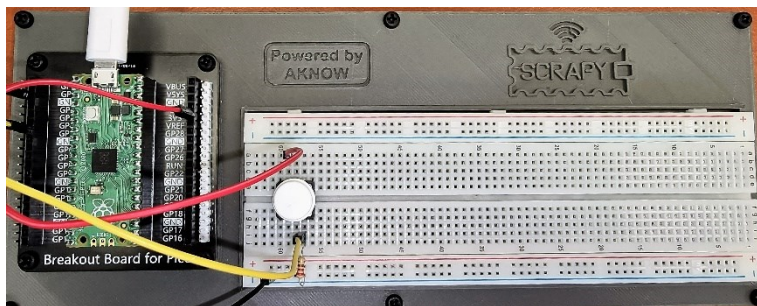
[button.py] x
1 from machine import Pin #import Pin and ADC from library machine
2 from time import sleep #import sleep from library time
3
4 button = Pin(5, Pin.IN, Pin.PULL_DOWN)
5
6 while True:
7     if button.value():
8         print("Button is pressed!")
9         sleep(1)
10

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνιση του κυκλώματος χρησιμοποιώντας το παρεχόμενο υλικό



### 3. Buzzer

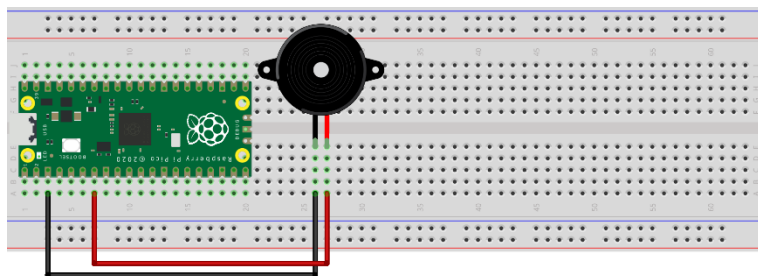
#### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε έναν βομβητή (Buzzer). Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `buzzer.py`. Έπειτα είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

#### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 2 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x Βομβητής

#### Διάγραμμα συνδεσμολογίας

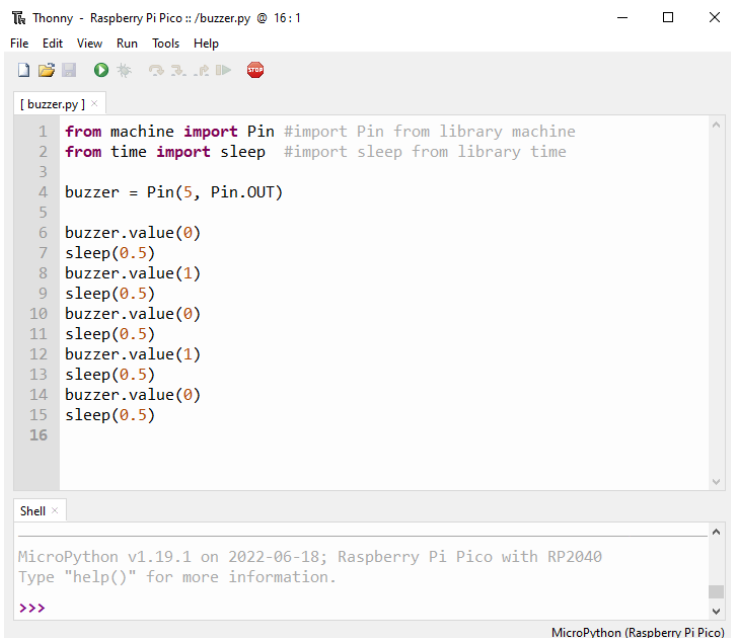


fritzing

- συνδέστε το μεγαλύτερο άκρο (+) του βομβητή στην ακίδα GPIO5
- συνδέστε το μικρότερο άκρο (-) του βομβητή σε μια ακίδα GND

## Κώδικας

Κώδικας MicroPython για το πείραμα:



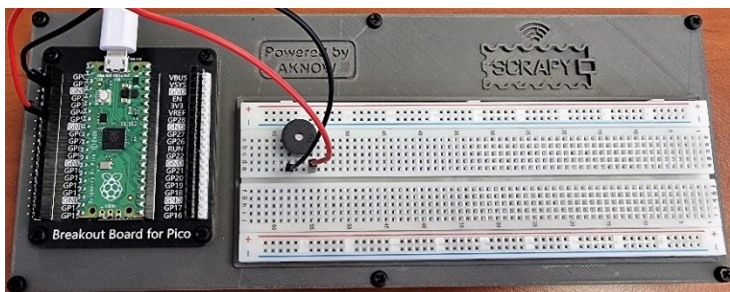
```

Thonny - Raspberry Pi Pico :: /buzzer.py © 16:1
File Edit View Run Tools Help
[ buzzer.py ] x
1 from machine import Pin #import Pin from library machine
2 from time import sleep #import sleep from library time
3
4 buzzer = Pin(5, Pin.OUT)
5
6 buzzer.value(0)
7 sleep(0.5)
8 buzzer.value(1)
9 sleep(0.5)
10 buzzer.value(0)
11 sleep(0.5)
12 buzzer.value(1)
13 sleep(0.5)
14 buzzer.value(0)
15 sleep(0.5)
16

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του κυκλώματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 4. Ποτενσιόμετρο

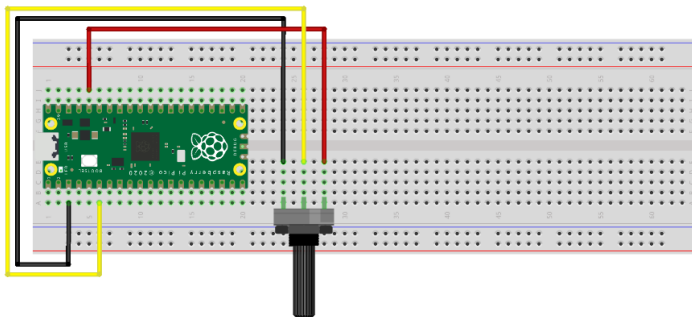
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε το ποτενσιόμετρο. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `pot.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x Ποτενσιόμετρο

### Διάγραμμα συνδεσμολογίας



fritzing

- Το μαύρο καλώδιο πρέπει να συνδεθεί στην ακίδα GND (Pin 3).
- Το κίτρινο καλώδιο πρέπει να συνδεθεί στην ακίδα GPIO5
- Το κόκκινο καλώδιο να συνδεθεί σε ακίδα τροφοδοσίας 3V3
- Γυρίστε το ποτενσιόμετρο προς τα αριστερά ώστε να είναι απενεργοποιημένο

## Κώδικας

### Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico :: /pot.py @ 9:1
File Edit View Run Tools Help

[ pot.py ] x
1 from machine import Pin, ADC #import Pin from library machine
2 from time import sleep #import sleep from library time
3
4 pot = ADC(Pin(26))
5
6 while True:
7     print(pot.read_u16())
8     sleep(0.5)
9

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του κυκλώματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 5. RGB LED

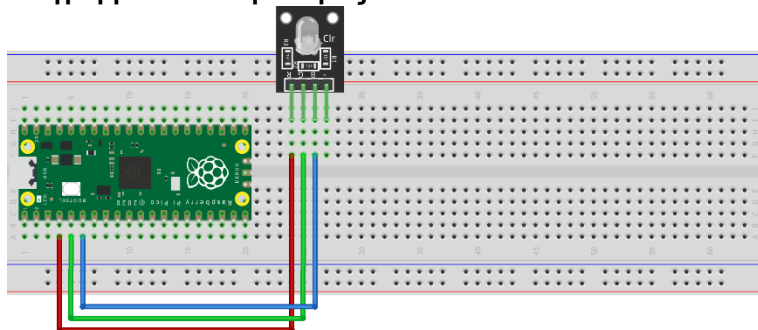
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε μια λυχνία μονάδας LED RGB. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `rgb.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 4 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x RGB μονάδα LED

### Διάγραμμα συνδεσμολογίας



fritzing

- Το κόκκινο καλώδιο πρέπει να συνδεθεί στο GPIO2
- Το πράσινο καλώδιο πρέπει να είναι συνδεδεμένο στο GPIO3
- Το μπλε καλώδιο πρέπει να συνδεθεί στο GPIO4
- Δεν απαιτείται σύνδεση GND

## Κώδικας

Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico :: /rgb.py © 14:1
File Edit View Run Tools Help

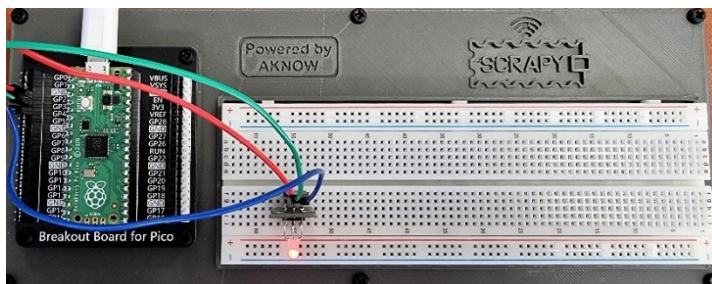
1 from machine import Pin, PWM #import Pin from library machine
2 from time import sleep #import sleep from library time
3 import random
4 Led_R = PWM(Pin(2))
5 Led_G = PWM(Pin(3))
6 Led_B = PWM(Pin(4))
7 # Define the frequency
8 Led_R.freq(2000)
9 Led_G.freq(2000)
10 Led_B.freq(2000)
11 while True:
12     # range of random numbers
13     R=random.randint(0,65535)
14     G=random.randint(0,65535)
15     B=random.randint(0,65535)
16     print(R,G,B)
17     Led_R.duty_u16(R)
18     Led_G.duty_u16(G)
19     Led_B.duty_u16(B)
20     sleep(1)

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του κυκλώματος χρησιμοποιώντας το παρεχόμενο υλικό:





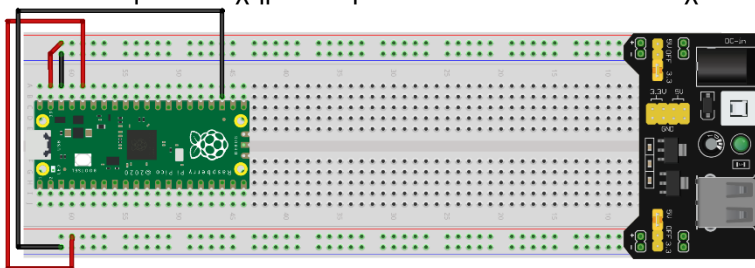
## Προχωρημένα πειράματα

Πριν προχωρήσουμε σε αυτήν την ενότητα και στην επόμενη, πρέπει να κάνουμε μερικές τροποποιήσεις στο SCRAPY Kit. Αυτό περιλαμβάνει την προσθήκη μερικών επιπλέον εξαρτημάτων και τη συνδεσιμότητά τους.

### Απαιτούμενο υλικό

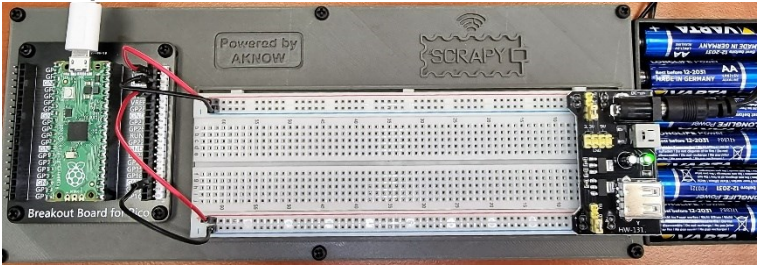
- Πακέτο μπαταριών 1 x 6-AA
- 1 x MB-102 μονάδα ισχύος
- 4 x καλώδια jumper αρσενικού σε αρσενικό

Ακολουθήστε το σχηματικό για να συνδέσετε τα νέα στοιχεία:



fritzing

## Κύκλωμα



- αυτή η ρύθμιση θα χρησιμοποιηθεί για τα υπόλοιπα πειράματα
- Επάνω πλαϊνές συνδέσεις: VSYS 5V ((+) κόκκινο) και GND ((-) μαύρο)
- Συνδέσεις κάτω πλευράς: 3V3 ((+) κόκκινο) και GND ((-) μαύρο)

## 6. LDR Φωτοαντίσταση

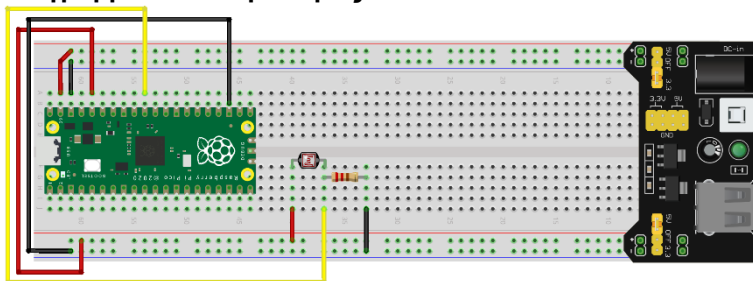
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε μια φωτοαντίσταση LDR. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `ldr.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x αντίσταση 220Ωm
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x φωτοαντίσταση LDR

### Διάγραμμα συνδεσμολογίας



fritzing

- Η αριστερή πλευρά του LDR είναι συνδεδεμένη στη γραμμή 3V ((+) κόκκινο)
- Η δεξιά πλευρά του LDR συνδέεται με μια αντίσταση 220 Ohm και στην ακίδα GPIO26 ADC (κίτρινο καλώδιο)
- η δεξιά πλευρά της αντίστασης συνδέεται με τη ράγα GND ((-) μαύρο)

## Κώδικας

### Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico ::/ldr.py @ 10: 1
File Edit View Run Tools Help

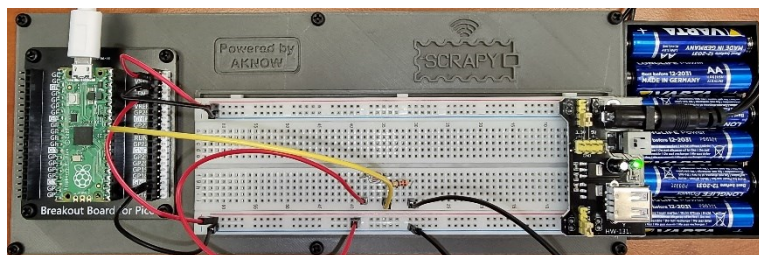
[ ldr.py ] x
1 from machine import Pin, ADC #import Pin and ADC from library machine
2 from time import sleep #import sleep from library time
3
4 ldr = ADC(Pin(26))
5
6
7 while True:
8     print(ldr.read_u16())
9     sleep(1)
10 |

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
8642
7329
4032
3504
3840
4040

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 7. Σερβοκινητήρας

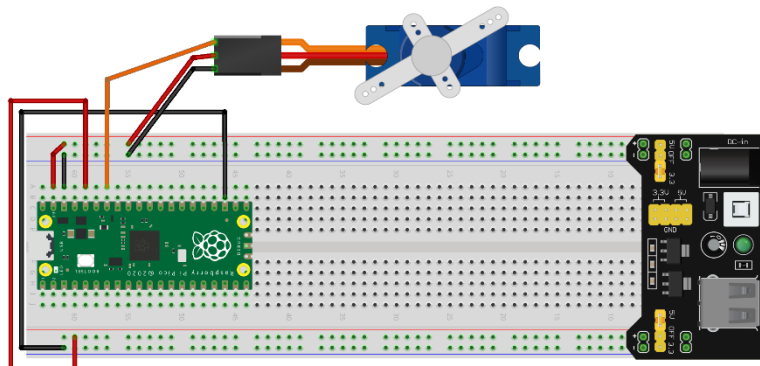
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε έναν σερβοκινητήρα. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `servo.py`. Έπειτα μπορείτε να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x SG-90 σερβοκινητήρα

### Διάγραμμα συνδεσμολογίας



fritzing

- το κόκκινο καλώδιο συνδέεται στη γραμμή 5V (+)
- Το μαύρο/καφέ καλώδιο συνδέεται στη γραμμή GND (-)
- Το πορτοκαλί καλώδιο είναι συνδεδεμένο στην ακίδα ADC GPIO28

## Κώδικας

Κώδικας MicroPython για το πείραμα:

```

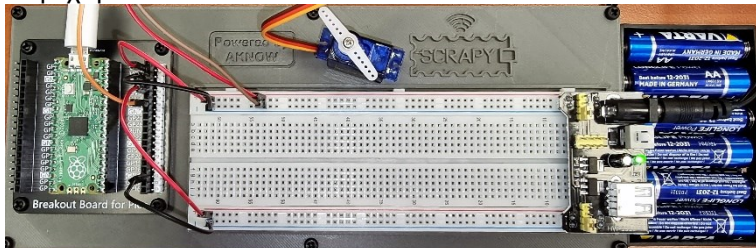
Thonny - Raspberry Pi Pico :: /servo.py @ 13:20
File Edit View Run Tools Help

[servo.py]
1 from machine import Pin, PWM #import Pin and PWM from library machine
2 from time import sleep #import sleep from library time
3
4 pwm = PWM(Pin(28))
5 pwm.freq(50)
6
7 while True:
8     for position in range(1000,9000,50):
9         pwm.duty_u16(position)
10        sleep(0.01)
11     for position in range(9000,1000,-50):
12         pwm.duty_u16(position)
13         sleep(0.01)

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 8. Οθόνη OLED I2C SSD1306

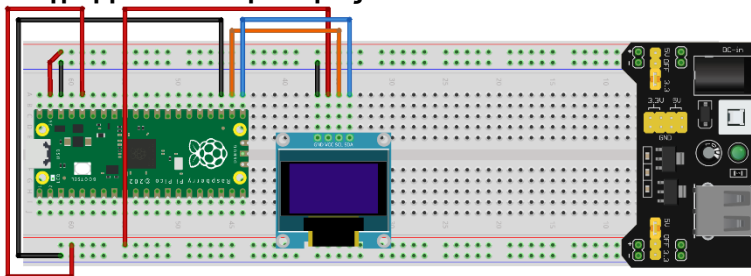
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε την οθόνη I2C ICC OLED. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `oled.py`. Έπειτα μπορείτε να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 4 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x OLED I2C SSD1306 οθόνη

### Διάγραμμα συνδεσμολογίας



fritzing

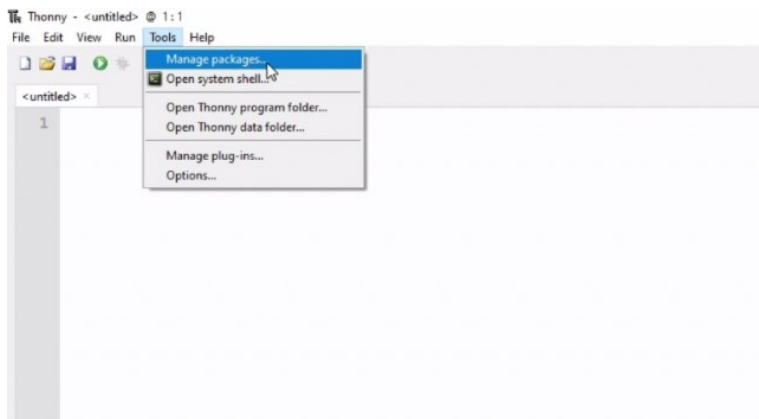
- το κόκκινο καλώδιο συνδέεται στη γραμμή 3V3 (+)
- Το μαύρο καλώδιο είναι συνδεδεμένο στη γραμμή GND (-)

- Το πορτοκαλί καλώδιο είναι συνδεδεμένο στην ακίδα GPIO17 I2C0 SCL
- Το μπλε καλώδιο συνδέεται στην ακίδα GPIO26 I2C0 SDA

## Κώδικας

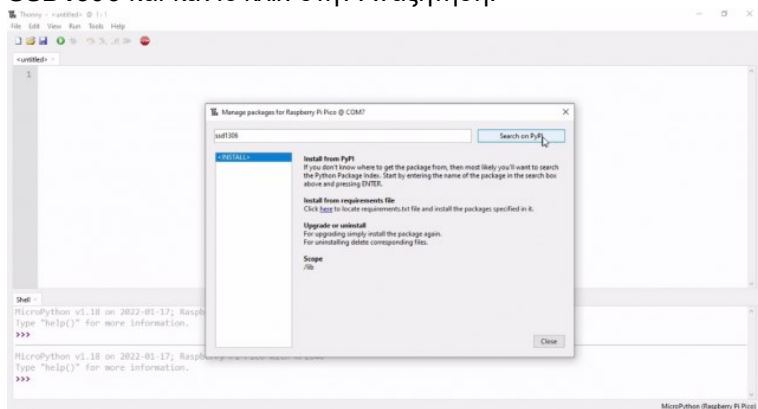
Πριν ξεκινήσουμε τον προγραμματισμό, την οθόνη OLED, πρέπει πρώτα να προσθέσουμε το πακέτο SSD1306 στο RPi Pico μας. Για να το κάνετε αυτό, ακολουθήστε τα παρακάτω βήματα:

1. Ανοίξτε το Thonny και μεταβείτε **στα Tools → Manage packages (Εργαλεία → Διαχείριση πακέτων...)**

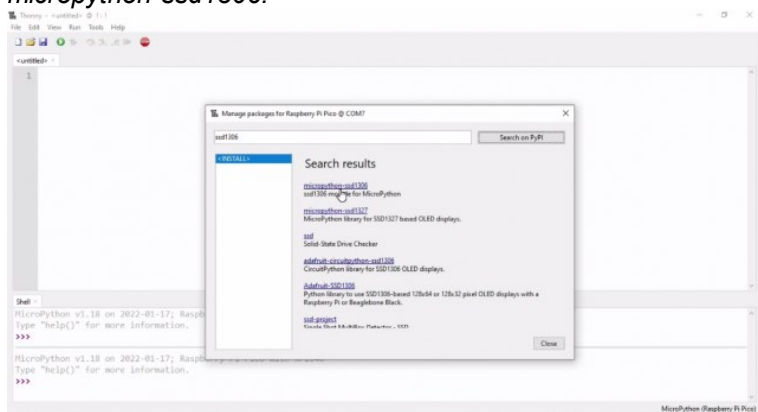




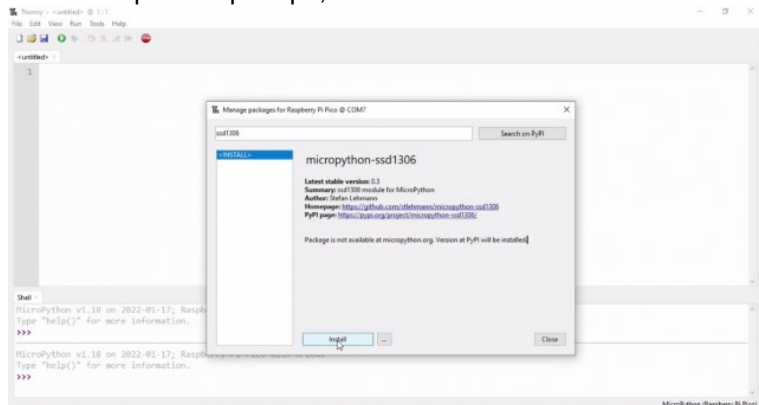
2. Στο παράθυρο διαχείρισης πακέτων, πληκτρολογήστε **SSD1306** και κάντε κλικ στην Αναζήτηση.



3. Μόλις ολοκληρωθεί η αναζήτηση, κάντε κλικ στο *micropython-ssd1306*.



#### 4. Στο επόμενο παράθυρο, κάντε κλικ στο *Install*.



5. Περιμένετε να εγκατασταθεί το πακέτο και μετά κάντε κλικ στο κλείσιμο.

Τώρα, είμαστε έτοιμοι να προχωρήσουμε στον προγραμματισμό της οθόνης OLED.

## Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico :: /oled.py @ 20: 1
File Edit View Run Tools Help

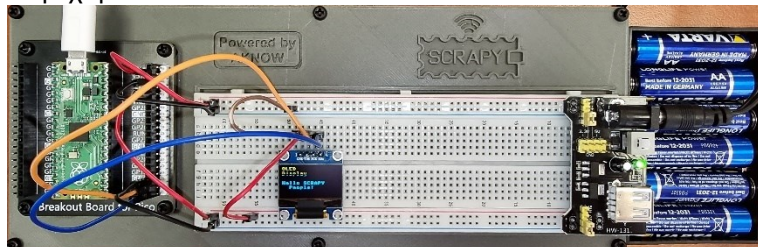
[oled.py] x
1 from machine import Pin, I2C
2 import ssd1306
3
4 WIDTH =128
5 HEIGHT = 64
6
7 PIN_SCL = 17
8 PIN_SDA = 16
9
10 i2c = I2C(0,scl=Pin(PIN_SCL),sda=Pin(PIN_SDA))
11 oled = ssd1306.SSD1306_I2C(WIDTH,HEIGHT,i2c)
12 oled.fill(0)
13
14 oled.text("OLED", 0, 0)
15 oled.text("Display", 0, 10)
16 oled.text("Hello SCRAPY", 0, 30)
17 oled.text(" People!", 0, 40)
18
19 oled.show()
20 |

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
>>>

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 9. Μονάδα Joystick

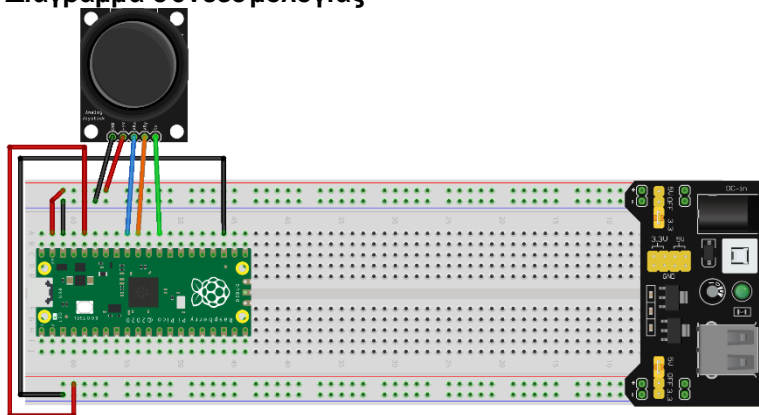
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε μια μονάδα joystick. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `joystick.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 5 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x μονάδα Joystick

### Διάγραμμα συνδεσμολογίας



fritzing

- Το +5V (κόκκινο καλώδιο) συνδέεται στη γραμμή 5V (+)
- Το GND (μαύρο καλώδιο) είναι συνδεδεμένο στη γραμμή GND (-)
- Το VRx (μπλε καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO27 ADC1

- Το VRy (πορτοκαλί καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO26 ADC0
- Το καλώδιο SW (πράσινο) είναι συνδεδεμένο στην ακίδα GPIO22

## Κώδικας

Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico :: /joystick.py @ 18: 5
File Edit View Run Tools Help

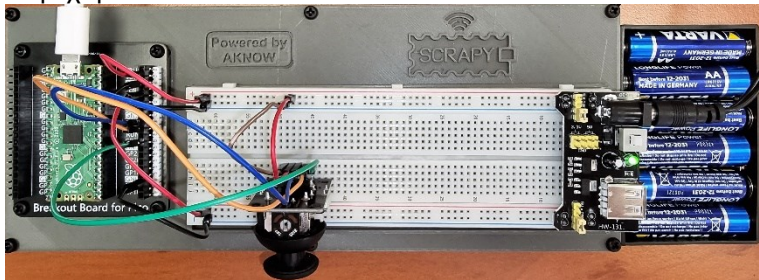
[joystick.py]
1 from machine import Pin, ADC
2 from time import sleep
3
4 VRX = ADC(Pin(27))
5 VRY = ADC(Pin(26))
6 SW = Pin(22, Pin.IN, Pin.PULL_UP)
7
8 while True:
9     xAxis = VRX.read_u16()
10    yAxis = VRY.read_u16()
11    switch = SW.value()
12
13    print("X-axis: " + str(xAxis) + ", Y-axis: " + str(yAxis) + ", Switch " + str(switch))
14    if switch == 0:
15        print("Push button pressed!")
16    print(" ")
17    sleep(1)

Shell
type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
X-axis: 32792, Y-axis: 34616, Switch 1
X-axis: 32471, Y-axis: 34520, Switch 1
X-axis: 32455, Y-axis: 34472, Switch 1

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## Πειράματα με αισθητήρες

### 10. Αισθητήρας σταγόνας βροχής

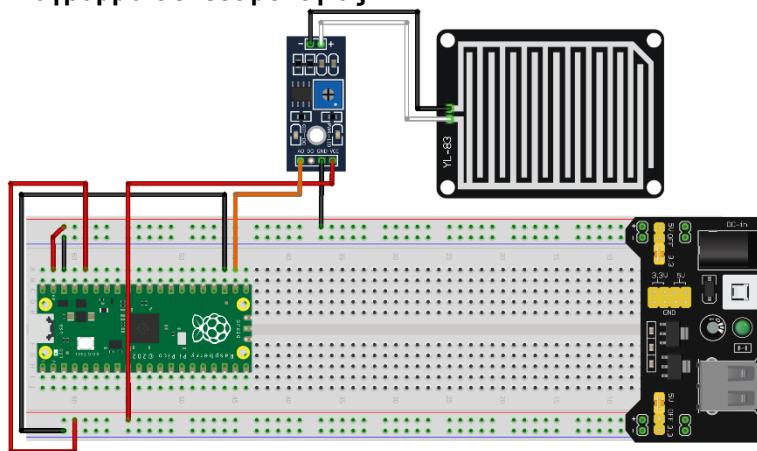
#### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα σταγόνας βροχής. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `raindrop.py`. Έπειτα μπορείτε να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

#### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x Αισθητήρας σταγόνας βροχής

#### Διάγραμμα συνδεσμολογίας

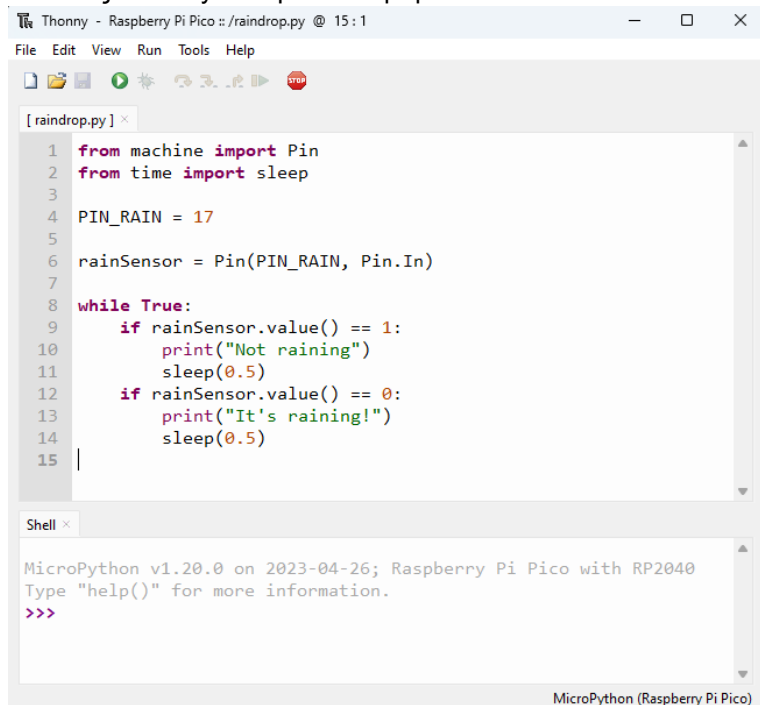


fritzing

- 3v3V (κόκκινο καλώδιο) συνδέεται με τη γραμμή 3v3V (+)
- Το GND (μαύρο καλώδιο) είναι συνδεδεμένο στη γραμμή GND (-)
- Το DO (πορτοκαλί καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO17

## Κώδικας

Κώδικας MicroPython για το πείραμα:



```

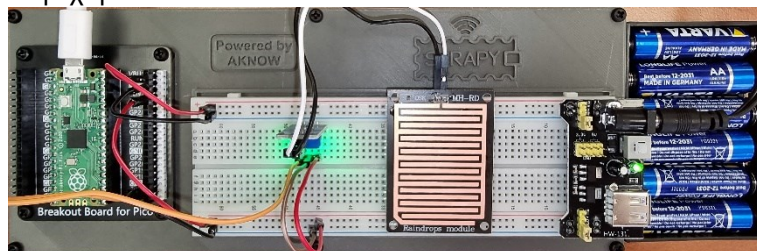
Thonny - Raspberry Pi Pico :: /raindrop.py @ 15:1
File Edit View Run Tools Help

[ raindrop.py ] x
1 from machine import Pin
2 from time import sleep
3
4 PIN_RAIN = 17
5
6 rainSensor = Pin(PIN_RAIN, Pin.In)
7
8 while True:
9     if rainSensor.value() == 1:
10        print("Not raining")
11        sleep(0.5)
12    if rainSensor.value() == 0:
13        print("It's raining!")
14        sleep(0.5)
15

Shell x
MicroPython v1.20.0 on 2023-04-26; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:





## 11. Αισθητήρας υπερήχων HC-SR04

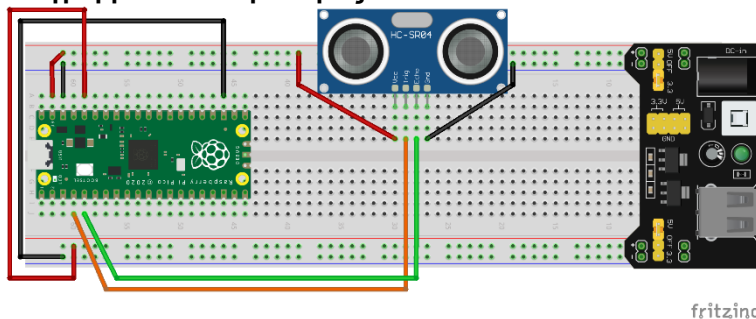
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα υπερήχων HC-SR04. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `ultrasonic.py`. Έπειτα μπορείτε να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x HC-SR04 Αισθητήρας υπερήχων

### Διάγραμμα συνδεσμολογίας

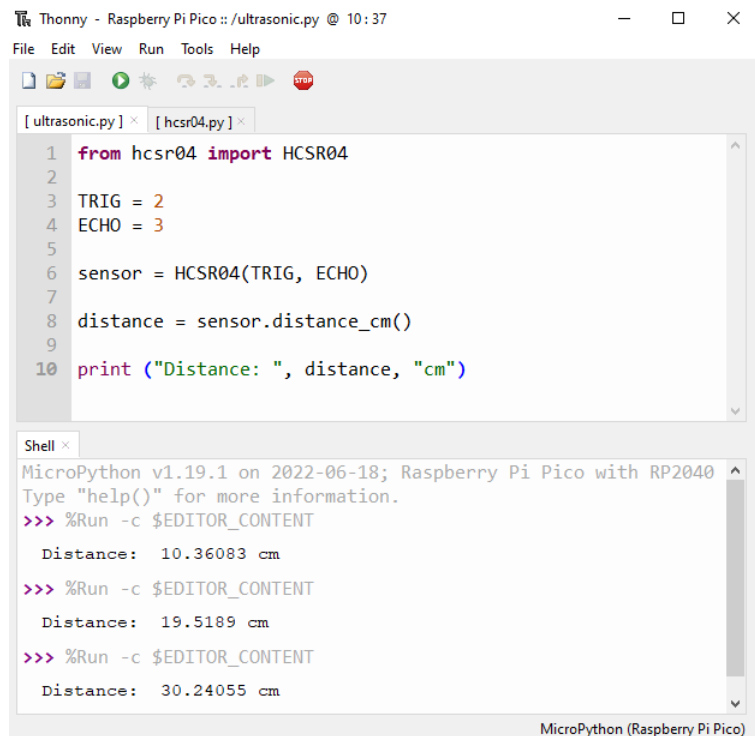


- Το VCC (κόκκινο καλώδιο) είναι συνδεδεμένο στη γραμμή 5V (+)
- GND (μαύρο καλώδιο) συνδέεται στη γραμμή GND (-)
- TRIG (πορτοκαλί καλώδιο) συνδέεται στην ακίδα GPIO2
- ECHO (πράσινο καλώδιο) συνδέεται στην ακίδα GPIO3.

## Κώδικας

Για να χρησιμοποιήσουμε τον αισθητήρα υπερήχων HC-SR04 μπορούμε να αναπτύξουμε το δικό μας πρόγραμμα ή να χρησιμοποιήσουμε μία από τις διαθέσιμες βιβλιοθήκες που υπάρχουν στο διαδίκτυο, π.χ. στο [Github](#). Εάν επιλέξετε να κάνετε λήψη της βιβλιοθήκης *hcsr04.py*, θα πρέπει να την αποθηκεύσετε στο Pico σας με το ίδιο όνομα.

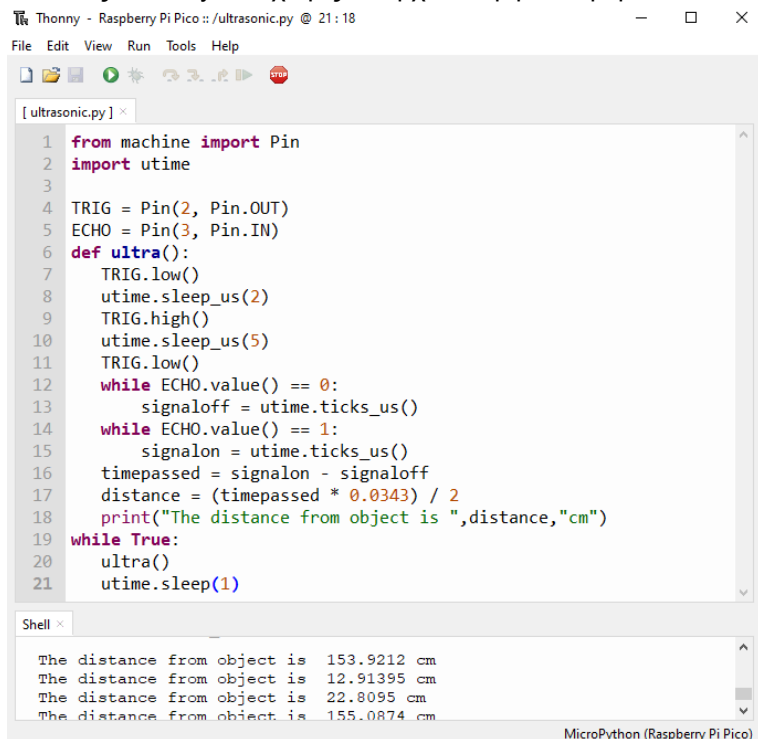
Κώδικας MicroPython χρησιμοποιώντας μια υπάρχουσα βιβλιοθήκη:



```
Thonny - Raspberry Pi Pico :: /ultrasonic.py @ 10:37
File Edit View Run Tools Help
[ ultrasonic.py ] x [ hcsr04.py ] x
1 from hcsr04 import HCSR04
2
3 TRIG = 2
4 ECHO = 3
5
6 sensor = HCSR04(TRIG, ECHO)
7
8 distance = sensor.distance_cm()
9
10 print ("Distance: ", distance, "cm")

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
    Distance: 10.36083 cm
>>> %Run -c $EDITOR_CONTENT
    Distance: 19.5189 cm
>>> %Run -c $EDITOR_CONTENT
    Distance: 30.24055 cm
MicroPython (Raspberry Pi Pico)
```

## Κώδικας MicroPython χωρίς υπάρχουσα βιβλιοθήκη:



```

1  from machine import Pin
2  import utime
3
4  TRIG = Pin(2, Pin.OUT)
5  ECHO = Pin(3, Pin.IN)
6  def ultra():
7      TRIG.low()
8      utime.sleep_us(2)
9      TRIG.high()
10     utime.sleep_us(5)
11     TRIG.low()
12     while ECHO.value() == 0:
13         signaloff = utime.ticks_us()
14     while ECHO.value() == 1:
15         signalon = utime.ticks_us()
16     timepassed = signalon - signaloff
17     distance = (timepassed * 0.0343) / 2
18     print("The distance from object is ",distance,"cm")
19     while True:
20         ultra()
21         utime.sleep(1)
    
```

Shell

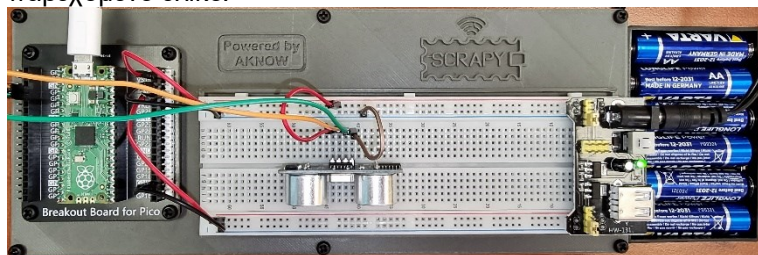
```

The distance from object is 153.9212 cm
The distance from object is 12.91395 cm
The distance from object is 22.8095 cm
The distance from object is 155.0874 cm
    
```

MicroPython (Raspberry Pi Pico)

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 12. Αισθητήρας κίνησης PIR

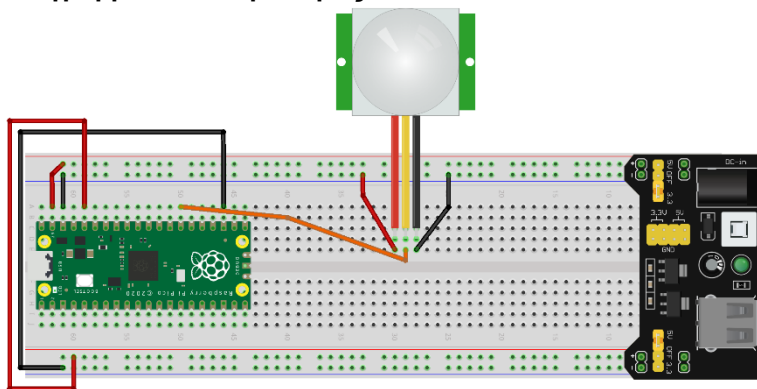
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα κίνησης PIR. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `motion.py`. Έπειτα μπορείτε να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper από αρσενικό σε θηλυκό
- 1 x αισθητήρας κίνησης PIR

### Διάγραμμα συνδεσμολογίας



fritzing

- Το VCC (κόκκινο καλώδιο) είναι συνδεδεμένο στη γραμμή 5V (+)
- GND (μαύρο καλώδιο) συνδέεται στη γραμμή GND (-)
- OUT (πορτοκαλί καλώδιο) συνδέεται στην ακίδα GPIO5

## Κώδικας

Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico :: /motion.py @ 6:12
File Edit View Run Tools Help

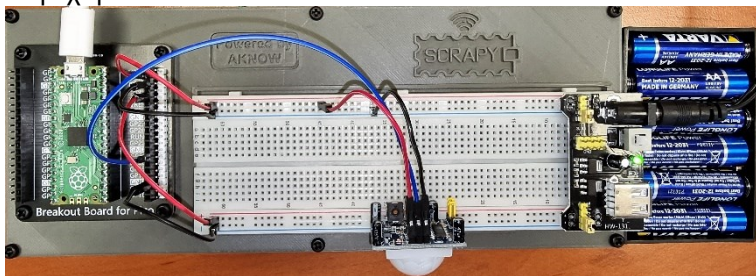
[ motion.py ] x
1 from machine import Pin
2 from time import sleep
3
4 PIR = Pin(21, Pin.IN, Pin.PULL_UP)
5
6 while True:
7     if PIR.value() == 1:
8         print("Motion detected!")
9         sleep(1)
10    if PIR.value() == 0:
11        print("No motion detected!")
12        sleep(1)

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
No motion detected!
No motion detected!
Motion detected!
Motion detected!

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 13. Αισθητήρας DHT11

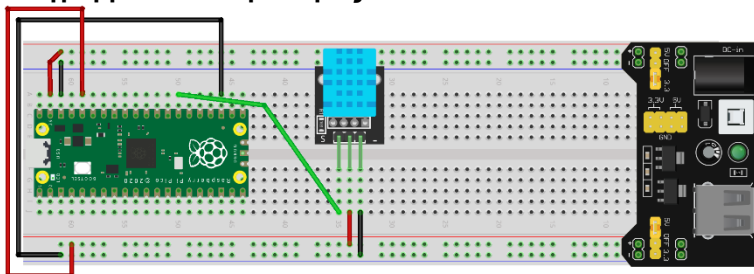
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα θερμοκρασίας και υγρασίας DHT11. Ανοίξτε το Θρογγυ Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `dht11.py`. Έπειτα μπορείτε να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x αισθητήρας θερμοκρασίας DHT11

### Διάγραμμα συνδεσμολογίας



fritzing

- Το VCC (κόκκινο καλώδιο) είναι συνδεδεμένο στη γραμμή 3V3 (+)
- Το GND (μαύρο καλώδιο) συνδέεται στη γραμμή GND (-)
- Το S (πράσινο καλώδιο) συνδέεται στην ακίδα GPIO5.

## Κώδικας

Κώδικας MicroPython για το πείραμα:

```

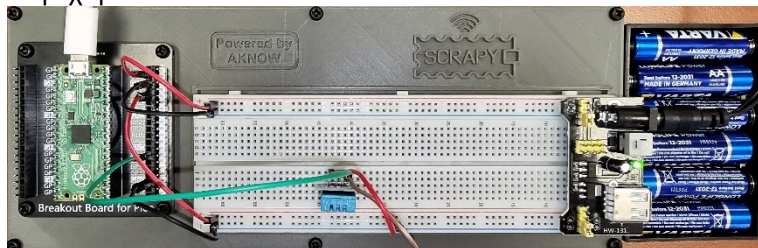
Thonny - Raspberry Pi Pico :: /dht11.py @ 12: 38
File Edit View Run Tools Help
[dht11.py] x
1 from machine import Pin
2 from time import sleep
3 import dht
4
5 sensor = dht.DHT11(Pin(21))
6
7 while True:
8     sensor.measure()
9     temp = sensor.temperature()
10    hum = sensor.humidity()
11    print("Room temperature: ",temp, "Celcius")
12    print("Room humidity: ",hum, "%")
13    sleep(5)

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Room temperature: 27 Celcius
Room humidity: 34 %

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 14. Αισθητήρας κραδασμών SW-420

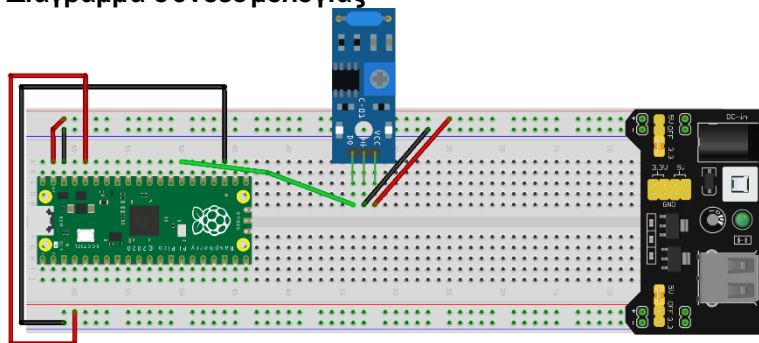
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα κραδασμών SW-420. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα vibration.py. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x αισθητήρας κραδασμών SW-420

### Διάγραμμα συνδεσμολογίας



fritzing

- Το VCC (κόκκινο καλώδιο) συνδέεται στη γραμμή 5V (+)
- Το GND (μαύρο καλώδιο) είναι συνδεδεμένο στη γραμμή GND (-)
- Το DO (πράσινο καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO21



## Κώδικας

Κώδικας MicroPython για το πείραμα:

```

Thonny - Raspberry Pi Pico :: /vibration.py @ 8:37
File Edit View Run Tools Help

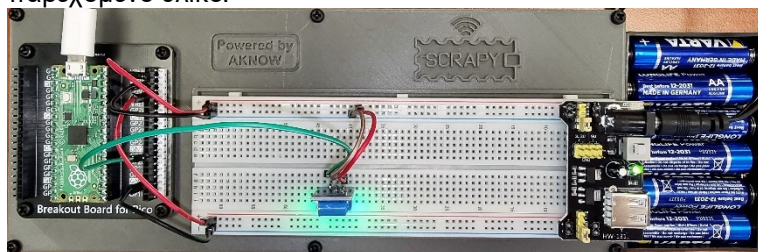
[vibration.py] x
1 from machine import Pin
2 from time import sleep
3
4 vibration = Pin(21, Pin.IN, Pin.PULL_UP)
5
6 while True:
7     if vibration.value() == 0:
8         print("Vibration detected!")

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Vibration detected!
Vibration detected!
Vibration detected!
Vibration detected!

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 15. Αισθητήρας φλόγας

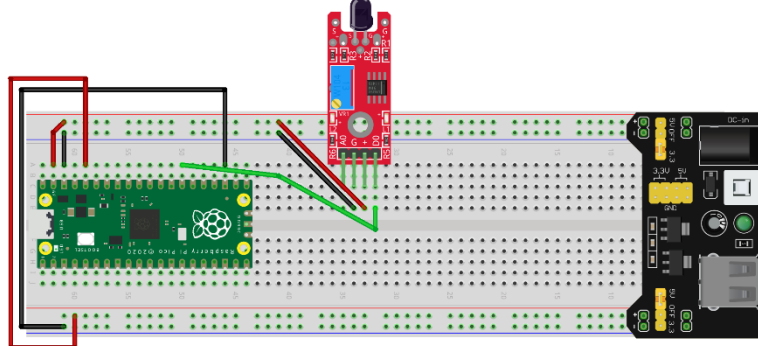
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα φλόγας KY-026. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `flame.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x KY-026 αισθητήρας φλόγας

### Διάγραμμα συνδεσμολογίας

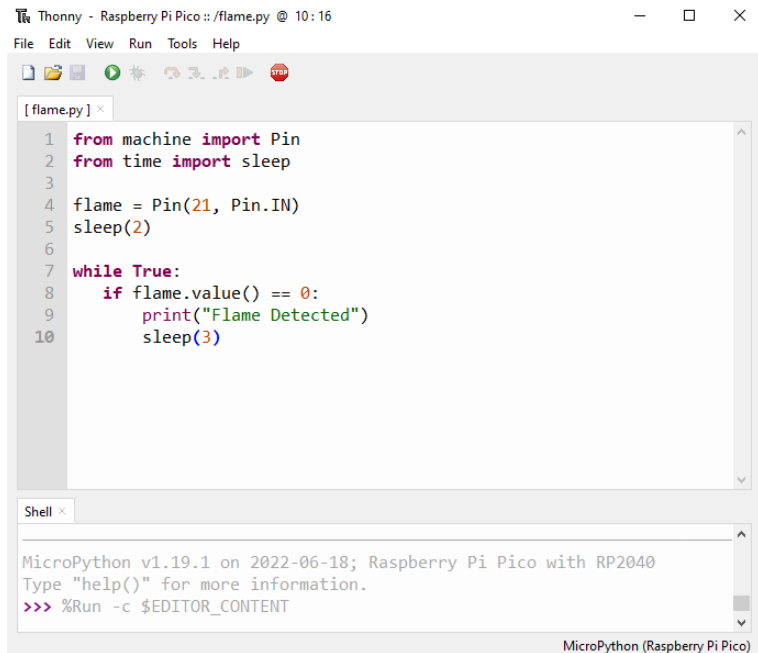


fritzing

- Το VCC (κόκκινο καλώδιο) συνδέεται στη γραμμή 5V (+)
- Το GND (μαύρο καλώδιο) είναι συνδεδεμένο στη γραμμή GND (-)
- Το DO (πράσινο καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO21

## Κώδικας

Κώδικας MicroPython για το πείραμα:



```

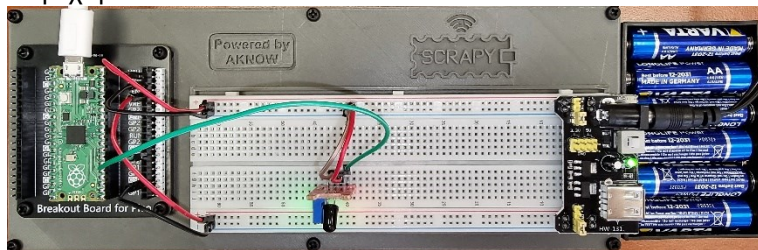
Thonny - Raspberry Pi Pico :: /flame.py @ 10:16
File Edit View Run Tools Help

[flame.py] x
1 from machine import Pin
2 from time import sleep
3
4 flame = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if flame.value() == 0:
9         print("Flame Detected")
10        sleep(3)

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 16. Αισθητήρας ανίχνευσης ήχου

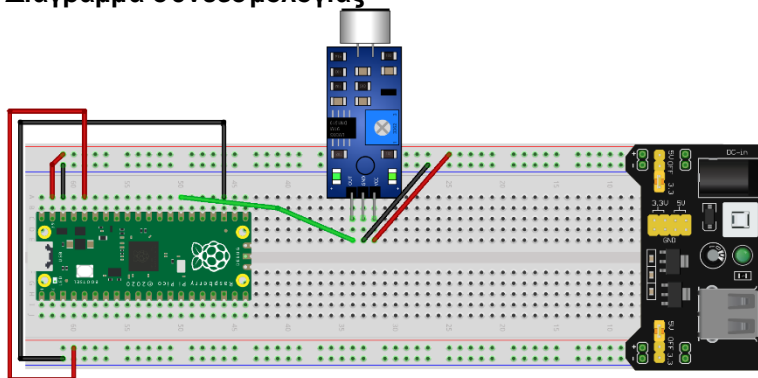
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα ανίχνευσης ήχου KY-037. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα `sound.py`. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x KY-037 αισθητήρας ήχου

### Διάγραμμα συνδεσμολογίας

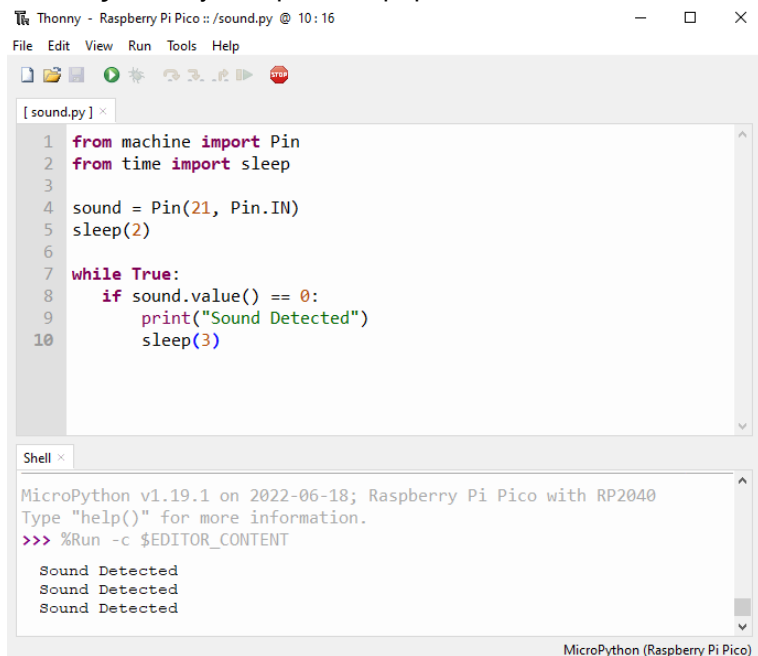


fritzing

- Το VCC (κόκκινο καλώδιο) συνδέεται στη γραμμή 5V (+)
- Το GND (μαύρο καλώδιο) είναι συνδεδεμένο στη γραμμή GND (-)
- Το DO/OUT (πράσινο καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO21

## Κώδικας

Κώδικας MicroPython για το πείραμα:



```

Thonny - Raspberry Pi Pico :: /sound.py @ 10:16
File Edit View Run Tools Help

[ sound.py ] x
1 from machine import Pin
2 from time import sleep
3
4 sound = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if sound.value() == 0:
9         print("Sound Detected")
10        sleep(3)

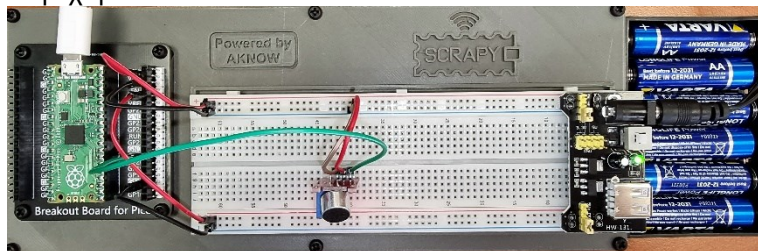
Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

Sound Detected
Sound Detected
Sound Detected

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 17. Αισθητήρας υγρασίας εδάφους

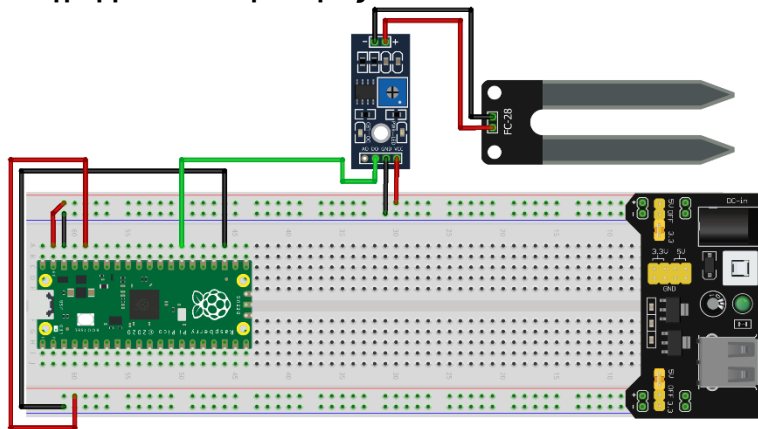
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα υγρασίας εδάφους. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα earth.py. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x KY-037 αισθητήρας ήχου

### Διάγραμμα συνδεσμολογίας



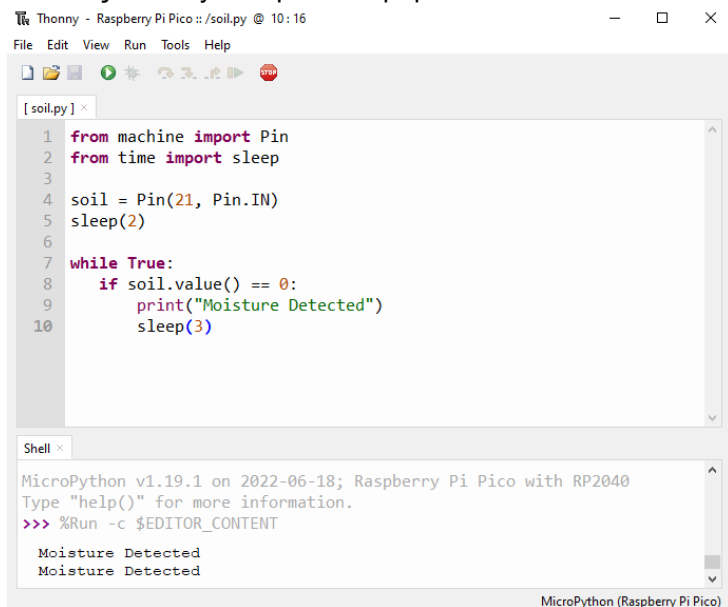
fritzing

- Το VCC (κόκκινο καλώδιο) συνδέεται στη γραμμή 5V (+)
- Το GND (μαύρο καλώδιο) είναι συνδεδεμένο στη γραμμή GND (-)

– Το DO/OUT (πράσινο καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO21

## Κώδικας

Κώδικας MicroPython για το πείραμα:



```

Thonny - Raspberry Pi Pico :: /soil.py @ 10:16
File Edit View Run Tools Help

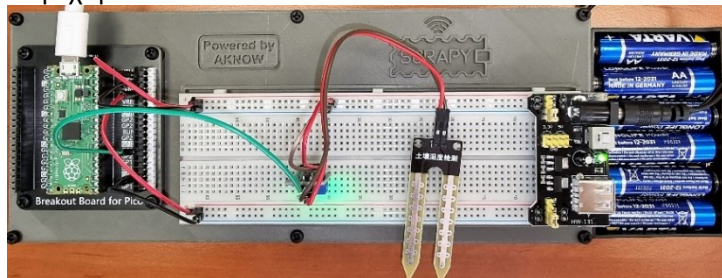
[soil.py] x
1 from machine import Pin
2 from time import sleep
3
4 soil = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if soil.value() == 0:
9         print("Moisture Detected")
10        sleep(3)

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Moisture Detected
Moisture Detected

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## 18. Αισθητήρας IR υπερέυθρων

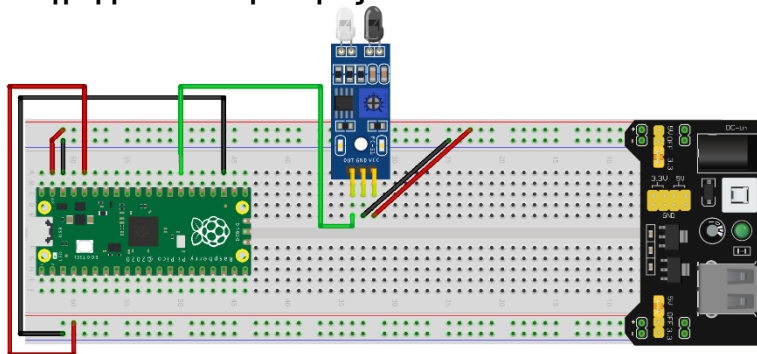
### Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα IR υπερέυθρων. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File → Save as..., (Αρχείο → Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα ir.py. Τότε είναι ώρα να συνδέσετε τα ηλεκτρονικά και να γράψετε το πρόγραμμά σας. Ακολουθήστε τις παρακάτω οδηγίες.

### Απαιτούμενο υλικό

- 1 x Raspberry Pi Pico
- 1 x breadboard πλήρους μεγέθους
- 1 x καλώδιο Micro-USB
- 1 x κιτ breadboard Pico
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x Αισθητήρας IR υπερέυθρων

### Διάγραμμα συνδεσμολογίας



fritzing

- Το VCC (κόκκινο καλώδιο) συνδέεται στη γραμμή 5V (+)
- Το GND (μαύρο καλώδιο) είναι συνδεδεμένο στη γραμμή GND (-)



– Το OUT (πράσινο καλώδιο) είναι συνδεδεμένο στην ακίδα GPIO21

## Κώδικας

Κώδικας MicroPython για το πείραμα:



```

Thonny - Raspberry Pi Pico :: /ir.py @ 13:17
File Edit View Run Tools Help

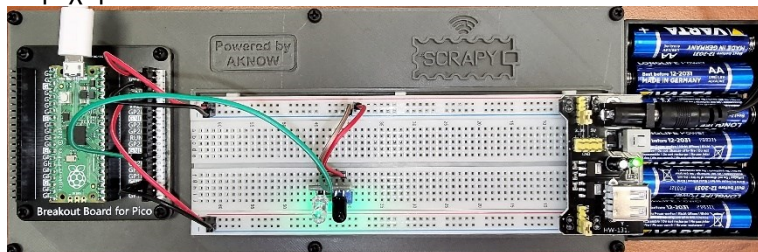
1 from machine import Pin
2 from time import sleep
3
4 ir = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if ir.value() == 1:
9         print("No obstacles")
10        sleep(3)
11    else:
12        print("Obstacle detected!")
13        sleep(3)
14

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Obstacle detected!
Obstacle detected!

MicroPython (Raspberry Pi Pico)
    
```

## Παράδειγμα

Εικόνα της εμφάνισης του πειράματος χρησιμοποιώντας το παρεχόμενο υλικό:



## ΠΑΡΑΡΤΗΜΑ: Συγκεντρωτικός Πίνακας MicroPython

Ψηφιακή έξοδος		
Κλήση της κλάσης ακίδας «Pin class»	<code>from machine import Pin</code>	
Ενεργοποίηση της ψηφιακής εξόδου του αντικειμένου.	<code>led = Pin(pin_value, Pin.OUT)</code>	pin_value από 0 έως 40
Άνοιγμα της ψηφιακής εξόδου (έξοδος 3.3V)	<code>led.value(1)</code>	ON (Ανοιχτή)
Κλείσιμο της ψηφιακής εξόδου (έξοδος 0V)	<code>led.value(0)</code>	OFF (Κλειστή)

Ψηφιακή είσοδος		
Κλήση της κλάσης ακίδας «Pin class»	<code>from machine import Pin</code>	
Ενεργοποίηση της ψηφιακής εξόδου του αντικειμένου.	<code>button = Pin(pin_value, Pin.IN)</code>	pin_value από 0 μέχρι 40
	<code>button = Pin(pin_value), Pin.IN, Pin.PULL_UP)</code>	Ενεργοποίηση αντίστασης PULL UP
	<code>button = Pin(pin_value), Pin.IN, Pin.PULL_DOWN)</code>	Ενεργοποίηση αντίστασης PULL DOWN
Ανάγνωση Εισόδου	<code>value = button.value(1)</code>	Η τιμή επιστροφής μπορεί να είναι 0 εάν η ακίδα είναι στα 0V ή 1 εάν η ακίδα είναι στα 3,3V

Αναλογική Έξοδος (Διαμόρφωση πλάτους παλμού - PWM)		
Κλήση της κλάσης PWM	<code>from machine import PWM</code>	
Ενεργοποίηση της αναλογικής εξόδου	<code>led = PWM(Pin(pin_value), frequency)</code>	pin_value από 0 μέχρι 40 frequency σε HZ, από 0 μέχρι 78125
Ανάγνωση εισόδου	<code>led.duty(duty_cycle)</code>	duty_cycle από 0 έως 1023 (έξοδος 0V σε έξοδο 3.3V αντίστοιχα)

Αναλογική είσοδος		
Κλήση της κλάσης ADC	<code>from machine import ADC</code>	
Ενεργοποίηση της αναλογικής εισόδου	<code>pot = ADC(Pin(pin_value))</code>	<code>pin_value</code> μπορεί να είναι GPIO26, GPIO27 και GPIO28
Δήλωση σχετικά με το σε ποια τάση θα δώσει η είσοδος τη μέγιστη τιμή της (σε ESP32 συνήθως είναι 3,3V)	<code>pot atten(ADC.ATTN_11DB)</code>	<b>ADC.ATTN_0DB:</b> τάση πλήρους εύρους: <b>1,2 V</b> <b>ADC.ATTN_2_5DB:</b> τάση πλήρους εύρους: <b>1,5 V</b> <b>ADC.ATTN_6DB:</b> τάση πλήρους εύρους: <b>2,0 V</b> <b>ADC.ATTN_11DB:</b> τάση πλήρους εύρους: <b>3,3 V</b>
Δήλωση του εύρους τιμών εισόδου (προεπιλογή 12 bit)	<code>pot.width(ADC.WIDTH_10BIT)</code>	<b>ADC.WIDTH_9BIT:</b> εύρος 0 έως 511 <b>ADC.WIDTH_10BIT:</b> εύρος 0 έως 1023 <b>ADC.WIDTH_11BIT:</b> εύρος 0 έως 2047 <b>ADC.WIDTH_12BIT:</b> εύρος 0 έως 4095
Ανάγνωση εισόδου	<code>value = pot.readl()</code>	<code>value</code> (η τιμή) είναι ένας ακέραιος αριθμός από το 0 έως το μέγιστο του εύρους που καθορίζεται από την εντολή <code>ADC.WIDTH_#BIT</code> (βλ. προηγούμενο)

Η βιβλιοθήκη του χρόνου		
Κλήση της κλάσης καθυστέρησης/ αναμονής «sleep»	<code>from time import sleep</code>	
Χρήση της κλάσης καθυστέρησης/ αναμονής «sleep»	<code>sleep(sec)</code>	<code>sec</code> είναι ο αριθμός των δευτερολέπτων για τα οποία θα καθυστερήσει το πρόγραμμα
Κλήση της κλάσης χρόνου	<code>import time</code>	
Χρήση της λειτουργίας χρόνου	<code>current_time = time()</code>	Η μεταβλητή <code>current_time</code> θα λάβει μια αριθμητική τιμή, ίση με τον αριθμό των δευτερολέπτων από την τελευταία επαναφορά στον πίνακα.

### Δομή Εντολής «If»

<pre>if &lt;expr1&gt;:     &lt;statement1&gt; elif &lt;expr2&gt;:     &lt;statement2&gt; elif &lt;expr3&gt;:     &lt;statement3&gt;     (...) else:     &lt;statementn&gt;</pre>	<p><b>&lt;expr#&gt;</b>: η συνθήκη ελέγχου που πρέπει να επιστρέψει True (Σωστό) ή False (Λάθος)</p> <p><b>&lt;statement#&gt;</b>: σύνολο εντολών που πρέπει να εκτελεστούν όταν ικανοποιείται η σχετική συνθήκη</p> <p><b>&lt;expr#&gt;</b> (e.g. το σύνολο &lt;statement2&gt;) εκτελείται όταν πληρείται η &lt;expr2&gt;)</p> <p><b>&lt;statementn&gt;</b>: σύνολο εντολών που εκτελούνται όταν δεν πληρείται καμία από τις συνθήκες &lt;expr#&gt;</p>
--	--

### Δομή της εντολής «While loop»

<pre>while &lt;expr&gt;:     &lt;statement(s)&gt;</pre>	<p><b>&lt;expr&gt;</b>: η συνθήκη ελέγχου που πρέπει να επιστρέψει True (Σωστό) ή False (Λάθος)</p> <p><b>&lt;statement#&gt;</b>: σύνολο εντολών που θα εκτελούνται εφόσον πληρείται η συνθήκη &lt;expr&gt;</p>
---	---

### Για τη δομή της εντολής «loop»

<pre>for &lt;var&gt; in &lt;iterable&gt;:     &lt;statement(s)&gt;</pre>	<p><b>&lt;iterable&gt;</b>: μια συλλογή στοιχείων, για παράδειγμα μια λίστα που περιέχει αριθμούς, αλφαριθμητικά κ.λπ.</p> <p><b>&lt;var&gt;</b>: μια μεταβλητή στην οποία εκχωρείται η τιμή του επόμενου στοιχείου της συλλογής &lt;iterable&gt;.</p> <p><b>&lt;statement(s)&gt;</b>: ένα σύνολο εντολών που εκτελούνται σε κάθε επανάληψη</p>
<pre>for &lt;var&gt; in range(&lt;start&gt;, &lt;end&gt;, &lt;step&gt;):     &lt;statement(s)&gt;</pre>	<p><b>range(&lt;start&gt;, &lt;end&gt;, &lt;step&gt;)</b>: συνάρτηση που επιστρέφει μια ακολουθία αριθμών από &lt;start&gt; έως &lt;end&gt;-1, με διαφορά &lt;step&gt; μεταξύ δύο διαδοχικών αριθμών (οι παράμετροι &lt;start&gt; και &lt;step&gt; είναι προαιρετικές).</p> <p><b>&lt;var&gt;</b>: μεταβλητή στην οποία αποδίδεται η τιμή του επόμενου στοιχείου της ακολουθίας που παράγεται από το εύρος.</p> <p><b>&lt;statement(s)&gt;</b>: ένα σύνολο εντολών που εκτελούνται σε κάθε επανάληψη</p>

### Διάφορα

DHT11	<pre>import dht</pre>	Εισαγωγή της βιβλιοθήκης DHT
	<pre>sensor = dht.DHT11(Pin(pin_number))</pre>	Ενεργοποίηση της μεταβλητής αισθητήρα με τον αντίστοιχο αριθμό ακίδας <code>pin_number</code> .
	<pre>sensor.measure()</pre>	Ενημέρωση των μεταβλητών του αισθητήρα
	<pre>temp = sensor.temperature()</pre>	Αποθήκευση τρέχουσας τιμής θερμοκρασίας
	<pre>hum = sensor.humidity()</pre>	Αποθήκευση τρέχουσας τιμής υγρασίας
ΟΘΟΝΗ OLED	<pre>from machine import I2C</pre>	Εισαγωγή της βιβλιοθήκης I2C
	<pre>import ssd1306</pre>	Εισαγωγή της βιβλιοθήκης <code>ssd1306</code>
	<pre>i2c = I2C(-1, scl=Pin(1), sda=Pin(0))</pre>	Ενεργοποίηση της μεταβλητής <code>i2c</code> στις ακίδες SCL & SDA του Pico
	<pre>oled_width = 128 oled_height = 64 oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)</pre>	Ενεργοποίηση της οθόνης
	<pre>oled.text('Hello, World 1!', 0, 0) oled.text('Hello, World 2!', 0, 10) oled.text('Hello, World 3!', 0, 20)</pre>	Αποθήκευση μηνυμάτων στο <code>buffer</code> οθόνης
	<pre>oled.show()</pre>	Εμφάνιση μηνυμάτων (απαραίτητο για την εμφάνιση των μηνυμάτων που είναι αποθηκευμένα στο <code>buffer</code> οθόνης)
	<pre>display.pixel(3, 4, 1)</pre>	Θέστε το <code>pixel</code> που βρίσκεται στη θέση (x,y) της οθόνης, με <code>x=3</code> & <code>y=4</code> , στην κατάσταση 1 (δηλ. εμφάνιση)



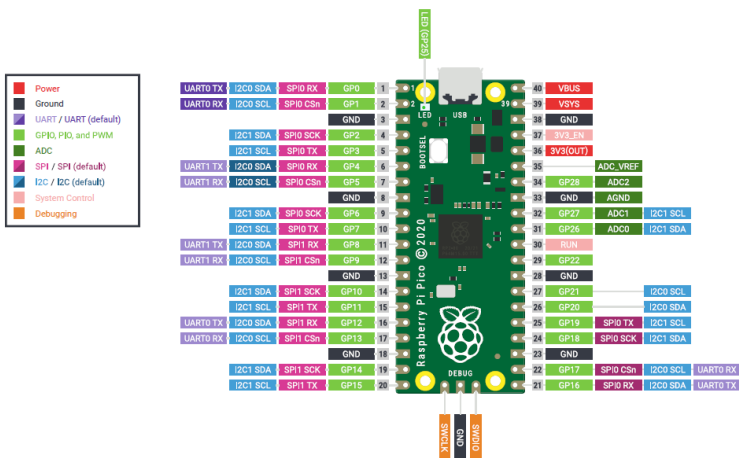
Co-funded by  
the European Union



2021-1-FR01-KA220-SCH-000031617

*Η υποστήριξη της Ευρωπαϊκής Επιτροπής στην παραγωγή της παρούσας έκδοσης δεν συνιστά αποδοχή του περιεχομένου, το οποίο αντικατοπτρίζει αποκλειστικά τις απόψεις των συντακτών, και η Επιτροπή δεν μπορεί να αναλάβει την ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν.*

## Raspberry Pi Pico Pinout



w: [scrapykit.eu](http://scrapykit.eu)  
e: [info@scrapykit.eu](mailto:info@scrapykit.eu)