

SmartHome4SENIORS Εγχειρίδιο του Kit





Co-funded by the European Union

Με τη χρηματοδότηση της Ευρωπαϊκής Ένωσης. Οι απόψεις και οι γνώμες που διατυπώνονται εκφράζουν αποκλειστικά τις απόψεις των συντακτών και δεν αντιπροσωπεύουν και'ανάγκη τις απόψεις της Ευρωπαϊκής Ένωσης ή του Ευρωπαϊκού Εκτελεστικού Οργανισμού Εκταίδευσης και Πολιτισμού (EACEA). Η Ευρωπαϊκή Ένωση και ο EACEA δεν μπορούν να θεωρηθούν υπεύθυνοι για τις εκφραζόμενες απόψεις.



ΠΡΟΕΙΔΟΠΟΙΗΣΗ

Όταν χρησιμοποιείτε αυτό το προϊόν, λάβετε υπόψη σας τα εξής:

- Αυτό το προϊόν περιέχει πολλά μικρά εξαρτήματα. Η κατάποση ή η κακή χρήση οποιουδήποτε από αυτά τα μέρη μπορεί να οδηγήσει σε σοβαρές ιατρικές επιπλοκές, συμπεριλαμβανομένου του θανάτου. Ζητήστε αμέσως ιατρική βοήθεια εάν συμβεί κάποιο ατύχημα.
- Η χρήση αυτού του προϊόντος και των εξαρτημάτων του κοντά σε πρίζες εναλλασσόμενου ρεύματος ή άλλα κυκλώματα απαγορεύεται αυστηρά, δεδομένου του πιθανού κινδύνου ηλεκτροπληξίας.
- Η χρήση αυτού του προϊόντος κοντά σε οποιοδήποτε υγρό ή φωτιά απαγορεύεται αυστηρά.
- 4. Κρατήστε αγώγιμα υλικά μακριά από αυτό το προϊόν.
- 5. Μην επιτρέπετε σε μικρά παιδιά να χρησιμοποιούν αυτό το προϊόν χωρίς την επίβλεψη ενηλίκου. Αυτό το προϊόν πρέπει να φυλάσσεται μακριά από παιδιά και κατοικίδια ζώα.
- 6. Μην αποθηκεύετε ή χρησιμοποιείτε αυτό το προϊόν σε οποιοδήποτε ακραίο περιβάλλον, όπως υπερβολική ζέστη ή κρύο, υψηλή υγρασία, κάτω από άμεσο ηλιακό φως κ.λπ.
- Να θυμάστε να διακόπτετε το κύκλωμα όταν δεν χρειάζεται.
- Ορισμένα μέρη αυτού του προϊόντος μπορεί να γίνουν ζεστά στην αφή όταν χρησιμοποιούνται σε ορισμένα σχέδια κυκλωμάτων, κάτι που είναι φυσιολογικό.
- Η ακατάλληλη χρήση μπορεί να προκαλέσει υπερθέρμανση.
- Η χρήση εξαρτημάτων που δεν είναι σύμφωνα με τις προδιαγραφές μπορεί να προκαλέσει ζημιά στο προϊόν.



Εισαγωγή

Το Κιτ Ηλεκτρονικών SmartHome4SENIORS είναι κατασκευασμένο με βάση τη χρήση του μικροελεγκτή Raspberry Pi Pico (RPi Pico). Το Κιτ δημιουργήθηκε στο πλαίσιο του ομώνυμου συγχρηματοδοτούμενου έργου Erasmus+ με αριθμό έργου 2021-1-DE02-KA220-ADU-000033587.

Η αυτοματοποίηση του έξυπνου σπιτιού είναι μια υπάρχουσα τάση που βοηθά χιλιάδες ανθρώπους να απλοποιήσουν τις διαδικασίες στα νοικοκυριά τους. Το SmartHome4SENIORS Kit σχεδιάστηκε και αναπτύχθηκε υπό το πρίσμα του χαμηλότερου επιπέδου ετοιμότητας των ηλικιωμένων να υιοθετήσουν και να κατανοήσουν τις τεχνολογικές λύσεις του έξυπνου σπιτιού.

Το Κιτ στοχεύει στην προώθηση της ασφαλούς και υγιούς διαβίωσης των ηλικιωμένων μέσω της πρακτικής εκπαιδευτικής μάθησης. Το Κιτ προσφέρει μια εξαιρετική ευκαιρία στους χρήστες να βουτήξουν στον κόσμο των DIY λύσεων έξυπνου οικιακού αυτοματισμού.

Το Κιτ περιλαμβάνει όλο το απαραίτητο υλικό (μικροελεγκτής, ηλεκτρονικά, αισθητήρες, περιφερειακά κ.λπ.) ενσωματώνοντας έννοιες φυσικού υπολογισμού και προγραμματισμού.

Το πεδίο εφαρμογής του παρόντος εγχειριδίου είναι:

- Να σας ενημερώσει σχετικά με τα εξαρτήματα του κιτ και τη χρήση των κύριων ηλεκτρονικών στοιχείων.
- Να σας καθοδηγήσει βήμα προς βήμα για την αποτελεσματική συναρμολόγηση του κιτ, λαμβάνοντας υπόψη τις σχετικές προφυλάξεις.
- Να σας παράσχει σεμινάρια για τη χρήση των εξαρτημάτων και τη σύνδεσή τους με τον μικροελεγκτή RPi Pico.
- Να σας παράσχει σεμινάρια με τις λειτουργίες και το πεδίο εφαρμογής κάθε ηλεκτρονικού στοιχείου.



Πίνακας Περιεχομένων

| Εισαγ | γωγή | 1 |
|-------------------|--|-------------|
| Πίνακ | κας Περιεχομένων | 2 |
| Περιλ | λαμβάνονται στο Κιτ SmartHome4SENIORS | 4 |
| Επεξι | ήγηση εξαρτημάτων | 6 |
| 1. | Τι είναι το breadboard (Πλακέτα Διασύνδεσης) ? | 6 |
| 2. | Τι είναι η αντίσταση | 7 |
| 3. | Τι είναι ένας πυκνωτής; | 10 |
| 4. | Τι είναι δίοδος; | 11 |
| 5. βρα | Τι είναι ένα καλώδιο jumper (γεφύρω αχυκυκλώματος); | οσης/ 12 |
| Προε ⁻ | τοιμασία Πειραμάτων | 13 |
| Συναβ | ρμολόγηση του Κιτ SmartHome | 22 |
| Smar | rtHome Kit Τοποθέτηση ηλεκτρονικών | 29 |
| Βασικ | κά πειράματα | 36 |
| 0. | "Hello SmartHome people!" | 36 |
| 1. | Ελέγξτε ένα LED | 38 |
| 2. | Πιέστε το κουμπί | 40 |
| 3. | Buzzer | 42 |
| 4. | Ποτενσιόμετρο | 44 |
| Προχ | (ωρημένα πειράματα | 46 |
| 5. | Μονάδα φωτεινών σηματοδοτών LED | 47 |
| 6. | LDR Φωτοαντίσταση | 50 |
| 7. | Κινητήρας συνεχούς ρεύματος (μικρός ανεμιστήρας |). 52 |
| 8. | SG-90 Σερβοκινητήρας | 54 |



| 9. | Οθόνη OLED I2C SSD130656 | | |
|---|-------------------------------------|--|--|
| 10. | Αναγνώστης RFID RC52261 | | |
| Πειράμ | ατα με αισθητήρες64 | | |
| 11. | Αισθητήρας σταγόνας βροχής64 | | |
| 12. | Αισθητήρας υπερήχων HC-SR0467 | | |
| 13. | Αισθητήρας κίνησης PIR70 | | |
| 14. | Αισθητήρας DHT1172 | | |
| 15. | Αισθητήρας φλόγας74 | | |
| 16. | MQ-135 Αισθητήρας ανίχνευσης αερίου | | |
| ΠΑΡΑΡΤΗΜΑ: Συγκεντρωτικός Πίνακας MicroPython | | | |

Απολαύστε την ανάγνωση και διασκεδάστε μέσω της πρακτικής εξάσκησης και του πειραματισμού με τοSmartHome4SENIORS Kit.



Περιλαμβάνονται στο Kιτ SmartHome4SENIORS



SmartHome 4SENIORS

2021-1-DE02-KA220-ADU-000033587





Επεξήγηση εξαρτημάτων

. . .

.

.

....

. . . .

.

1. Τι είναι το (Πλακέτα breadboard Διασύνδεσης) ?

. . .

. . .

....

To 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 πλήν (-) και παρέχει τη γειώση.

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 | | 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 | 1000000 | ± 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 | 4-Ζώνες | 5- Ζώνες | Κοινή Χρήση | |
|----------------------|------------------------------------|---|-----------------------------------|--|
| αντιστασης | Κωδικος χρώματος | Κωδικός χρώματος | | |
| 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 Εικόνας: Future) διαθέσιμο στη διεύθυνση https://www.tomshardware.com/how-to/resistor-color-codes

- Ο πρώτος σημαντικός αριθμός είναι κόκκινος και χρησιμοποιώντας τον αποκωδικοποιητή μπορούμε να δούμε ότι το κόκκινο έχει τιμή 2.
- Ο δεύτερος σημαντικός αριθμός είναι επίσης κόκκινος, οπότε μας δίνει 22.
- Ο πολλαπλασιαστής είναι καφέ και αυτός αποκωδικοποιείται στο 10. Αν πολλαπλασιάσουμε το 22 με το 10, θα έχουμε 220.
- 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.





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

- 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.
- Ο πολλαπλασιαστής είναι κόκκινος, και αυτός αποκωδικοποιείται στο 100. Αν πολλαπλασιάσουμε το 10 με το 100 θα έχουμε 1000.

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 100uF 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 n συμπεριφορά μονής κατεύθυνσης ονομάζεται αποκατάσταση και μετατροπή του εναλλασσόμενου χρησιμοποιείται τη για 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)

To 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 Ρίco σας. Κάντε κλικ στο κουμπί Εγκατάσταση για να αντιγράψετε το υλικολογισμικό στο Ρίco σας.

| 🙀 Install MicroPython firmware for Raspberry Pi Pico | × | | |
|---|--------|--|--|
| | | | |
| Here you can install or update MicroPython firmware on Raspberry Pi Pico. | | | |
| Plug in your Pico while holding the BOOTSEL button. Wait until device information appears. Click 'Install'. | | | |
| When the process finishes, your Pico will be running the latest version of MicroPython. Close the dialog and start programming! | | | |
| | | | |
| Version to be installed: v1.19.1 (2022-06-18) | | | |
| Target device location: D:\ | | | |
| Target device model: Raspberry Pi RP2 | | | |
| | | | |
| Install | Cancel | | |

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

| 🙀 Install MicroPython firmware for Raspberry Pi Pico | х | | |
|--|---|--|--|
| Here you can install or update MicroPython firmware on Raspberry Pi Pico. 1. Plug in your Pico while holding the BOOTSEL button. 2. Wait until device information appears. 3. Click 'Install'. When the process finishes, your Pico will be running the latest version of MicroPython. Close the dialog and start programming! | | | |
| Version to be installed: v1.19.1 (2022-06-18) Target device location: D:\ Target device model: Raspberry Pi RP2 Done! Install Close | | | |



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 διερμηνέα 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.
    v

    MicroPython (Raspberry Pi Pico)
    v
```

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 PiPico)
V
```



Η MicroPython σάς 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)
```

```
Shelt ×
>>> 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(9)
>>> led.value(9)
MicroPython (Raspberry PiPico)
```



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 αποθηκεύσετε 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.



| 隔 Thonny - <untitled> @ 4:13 File Edit View Run Tools Help</untitled> | _ | | × |
|--|------------|-----------|-------|
| | | | |
| <pre>1 from machine import Pin 2 led = Pin(25, Pin.OUT) 3 4 led.toggle() </pre> <pre> The Where to save to? × </pre> | | | ^ |
| This computer Raspberry Pi Pico | | | ~ |
| Shell < | | | < |
| MicroP | ython (Ras | pberry Pi | Pico) |

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.

| File name: | blink.py | ОК | Cancel |
|------------|----------|----|--------|
| | | | |

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 του Κιτ SmartHome

Η συναρμολόγηση απαιτεί τα ακόλουθα εξαρτήματα:

- 6 x Τεμάχια κόντρα πλακέ
- 10 x Μεταλλικές βίδες
- 10 x Μεταλλικά παξιμάδια



Η διαδικασία συναρμολόγησης είναι απλή και μπορεί να ολοκληρωθεί σε 7 βήματα. Όλα τα τεμάχια κόντρα πλακέ σημειώνονται στη δεξιά γωνία με την ακόλουθη ένδειξη:

- **ΒΡ**: Τεμάχιο βάσης
- Β: Τεμάχιο πίσω πλευράς
- **F**: Κομμάτι στην μπροστινή πλευρά
- L: Τεμάχιο αριστερής πλευράς
- **R**: Τεμάχιο δεξιάς πλευράς



Βήμα 1: Τοποθετήστε τη βάση από κόντρα πλακέ σε μια οριζόντια επιφάνεια.



Βήμα 2: Τοποθετήστε το κομμάτι της πίσω πλευράς όπως φαίνεται στην εικόνα.





Βήμα 3: Τοποθετήστε το μπροστινό κομμάτι όπως φαίνεται στην εικόνα.





Βήμα 4: Τοποθετήστε το κομμάτι κόντρα πλακέ της αριστερής πλευράς όπως φαίνεται στην εικόνα.





Βήμα 5: Τοποθετήστε το κομμάτι κόντρα πλακέ της δεξιάς πλευράς όπως φαίνεται στην εικόνα.





Βήμα 6: Σφίξτε τα κομμάτια κόντρα πλακέ μεταξύ τους χρησιμοποιώντας 10 βίδες και 10 παξιμάδια.

- 6 βίδες και παξιμάδια τοποθετούνται κάτω από το κομμάτι βάσης και είναι υπεύθυνα για τη συγκράτηση των τοίχων.
- 4 βίδες και παξιμάδια τοποθετούνται στα τοιχώματα της αριστερής και της δεξιάς πλευράς και είναι υπεύθυνα για τη σταθερότητα της κατασκευής.





Βήμα 7: Τοποθετήστε το κομμάτι της στέγης από κόντρα πλακέ πάνω στο μοντέλο του σπιτιού, όπως φαίνεται στην εικόνα:





SmartHome Kit Τοποθέτηση ηλεκτρονικών

Υπάρχουν διάφορα ηλεκτρονικά εξαρτήματα που πρέπει να τοποθετηθούν στο μοντέλο σπιτιού SmartHome. Οι παρακάτω εικόνες παρουσιάζουν λεπτομερώς ποια ηλεκτρονικά στοιχεία πρέπει να τοποθετηθούν και πώς να τοποθετηθούν στις διάφορες υποδοχές. Θα χρειαστείτε επίσης ένα σταυροκατσάβιδο και μια πένσα.



Τοποθετήστε το Raspberry Pi Pico στην πλακέτα breadboard, μαζί με τη μονάδα τροφοδοσίας (MB-102) και το κουμπί. Το breadboard έχει μια αυτοκόλλητη επιφάνεια από κάτω, οπότε



φροντίστε να αφαιρέσετε την προστατευτική στρώση και στη συνέχεια να το τοποθετήσετε στο βασικό κομμάτι του σπιτιού.

Υπάρχουν ορισμένα ηλεκτρονικά εξαρτήματα που πρέπει να τοποθετηθούν στον τοίχο της πίσω πλευράς. Αυτά είναι ο αισθητήρας φλόγας, ο αισθητήρας ποιότητας αέρα και ο αισθητήρας θερμοκρασίας και υγρασίας DHT11. Πρέπει να χρησιμοποιήσετε τα ακόλουθα αντικείμενα:

- Αισθητήρας φλόγας: 1 x βίδα και 1 x παξιμάδι.
 Τοποθετήστε το μέσω του μεσαίου τμήματος του αισθητήρα και βεβαιωθείτε ότι ο αισθητήρας (μαύρο μέρος) κοιτάζει προς τα πάνω.
- Αισθητήρας ποιότητας αέρα: 2 x βίδες και 2 x παξιμάδια. Τοποθετήστε το μέσω των οπών τοποθέτησης πάνω αριστερά και πάνω δεξιά.
- DHT11: 2 x βίδες και 2 x παξιμάδια. Πρέπει να λυγίσετε τον αισθητήρα (μπλε μέρος) προς τα εμπρός για να αποκτήσετε πρόσβαση στις οπές τοποθέτησης. Στη συνέχεια, τοποθετήστε τον αισθητήρα μέσω των οπών τοποθέτησης επάνω αριστερά και επάνω δεξιά.

Στο αριστερό τμήμα θα χρειαστεί να τοποθετήσετε έναν ανεμιστήρα συνεχούς ρεύματος και ένα φως LED. Για τον ανεμιστήρα DC θα χρησιμοποιήσετε τις βίδες και τα παξιμάδια που περιέχονται στη συσκευασία του (4 x βίδες και 4 x παξιμάδια). Τοποθετήστε το LED στην οπή τοποθέτησης και η πρόσφυση θα το κρατήσει στη θέση του.





Στο κομμάτι της δεξιάς πλευράς θα χρειαστεί να τοποθετήσετε έναν βομβητή 5V, ένα ποτενσιόμετρο και μια λυχνία LED.

Τοποθετήστε τον **βομβητή** στην επάνω δεξιά γωνιακή υποδοχή (βεβαιωθείτε ότι οι ακίδες του βρίσκονται στο εσωτερικό του μοντέλου σπιτιού) και η πρόσφυση θα τον κρατήσει στη θέση του.

Στο ποτενσιόμετρο έχει ήδη τοποθετηθεί ένας δακτύλιος με παξιμάδι, τον οποίο θα πρέπει να ξεβιδώσετε. Τοποθετήστε το ποτενσιόμετρο από το εσωτερικό του σπιτιού προς τα έξω, ώστε ο μοχλός να μοιάζει όπως στην παρακάτω εικόνα. Στη συνέχεια, βιδώστε πίσω τον δακτύλιο του παξιμαδιού μέχρι να σφίξει αρκετά για να το κρατήσει στη θέση του όταν γυρίζετε τον μοχλό.

Τέλος, πρέπει να εγκατασταθεί μια λυχνία LED. Τοποθετήστε το LED στην οπή τοποθέτησης και η τριβή θα το κρατήσει στη θέση του.







Υπάρχουν αρκετοί αισθητήρες και ηλεκτρονικά συστήματα που πρέπει να τοποθετηθούν στην μπροστινή πλευρά του μοντέλου του σπιτιού.

- Σερβοκινητήρας: 4 x βίδες. Τοποθετήστε τα από το εσωτερικό του σπιτιού προς το εξωτερικό και τοποθετήστε τα με τις βίδες που περιλαμβάνονται στη συσκευασία.
- PIR Αισθητήρας κίνησης: 2 x βίδες και 2 x παξιμάδια.
 Τοποθετήστε το μέσω των οπών τοποθέτησης πάνω αριστερά και κάτω δεξιά.
- RFID αισθητήρας: 2 x βίδες και 2 x παξιμάδια.
 Τοποθετήστε το μέσω των οπών τοποθέτησης πάνω αριστερά και κάτω δεξιά.
- OLED οθόνη: 2 x βίδες και 2 x παξιμάδια. Τοποθετήστε το μέσω των οπών τοποθέτησης πάνω αριστερά και κάτω δεξιά.
- Μονάδα φωτεινών σηματοδοτών: 2 x βίδες και 2 x παξιμάδια. Τοποθετήστε τη μονάδα από το εσωτερικό του σπιτιού προς τα έξω. Στη συνέχεια, τοποθετήστε την μέσω των δύο οπών τοποθέτησης στη δεξιά πλευρά.
- Αισθητήρας υπερήχων: Τοποθετήστε το από το εσωτερικό του σπιτιού προς τα έξω. Η πρόσφυση θα το κρατήσει στη θέση του.
- LED: Τοποθετήστε το LED στην οπή τοποθέτησης και η πρόσφυση θα το κρατήσει στη θέση του.






Τέλος, ο αισθητήρας βροχής πρέπει να εγκατασταθεί στην οροφή. Χρησιμοποιήστε 2 x βίδες και 2 x παξιμάδια και τοποθετήστε τον μέσω των οπών τοποθέτησης στην πάνω αριστερή και κάτω δεξιά πλευρά.



Αφού τοποθετήσετε όλα τα ηλεκτρονικά και τους αισθητήρες στη θέση τους, θα πρέπει να τα καλωδιώσετε χρησιμοποιώντας τα καλώδια Jumper που περιλαμβάνονται στη συσκευασία. Ωστόσο, αυτή η διαδικασία θα εξηγηθεί στην ενότητα με τα πειράματα, όπου θα εξηγηθούν οι διαφορετικές ακίδες GPIO του Raspberry Pi Pico και οι ακίδες κάθε ηλεκτρονικού εξαρτήματος ή αισθητήρα.

Στη συσκευασία θα βρείτε επίσης 6 μπαταρίες AA και μια θήκη μπαταριών. Τοποθετήστε τις μπαταρίες στη θήκη μπαταριών και κρατήστε την στην άκρη προς το παρόν. Σε ένα μεταγενέστερο tutorial θα μάθετε πού θα πρέπει να την τοποθετήσετε και πώς θα πρέπει να συνδεθεί με τη μονάδα τροφοδοσίας MB-102.



Βασικά πειράματα

0. "Hello SmartHome people!"

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

Η λειτουργία print()

 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.

| 1 💕 | 📕 💽 🎋 🕾 3et 🕨 🚥 |
|----------|---|
| hello.py | \star $	imes$ |
| 1 | <pre>print("Hello SmartHome people!")</pre> |

2. Ελέγξτε το παράθυρο shell.



 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.

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 ×

>>> %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 SmartHome people!" χρησιμοποιώντας μόνο μία λειτουργία print().



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 \rightarrow Save as..., (Αρχείο \rightarrow Αποθήκευση ως) επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα led.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 Micro-USB καλώδιο
- 2 x καλώδιο jumper αρσενικό σε αρσενικό
- 1 x breadboard πλήρους 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

2021-1-DE02-KA220-ADU-000033587



Κώδικας

Κωδικός MicroPython για το πείραμα

```
\times
Thonny - Raspberry Pi Pico :: /led.py @ 9:1
File Edit View Run Tools Help
🗋 💕 📓 🔹 🕸 👁 🕉 🖈 🖿 🚥
 [led.py] ×
      from machine import Pin #import Pin from library machine
   2
      from time import sleep #import sleep from library time
      led = Pin(5, Pin.OUT)
      while True:
   6
          led.toggle()
          sleep(1)
   9
 Shell >
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
 >>>
                                                     MicroPython (Raspberry Pi Pico)
```

Παράδειγμα





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. 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

Διάγραμμα συνδεσμολογίας

- 3 x καλώδια jumper αρσενικού σε αρσενικό
- 1 x καλώδιο Micro-USB
- 1 x κουμπί

1 x Αντίσταση 220 Ohm
1 x Καπάκι κουμπιού

fritzing

συνδέστε την πάνω αριστερή πλευρά του κουμπιού στον ακροδέκτη
 3v3 (κόκκινο καλώδιο)

συνδέστε την κάτω δεξιά πλευρά του κουμπιού στο GPIO5 (κίτρινο καλώδιο)

- συνδέστε έναν πείρο GND στη ράγα (-) (μαύρο καλώδιο)

 συνδέστε την αντίσταση 220 Ohm στη ράγα (-) και στην κάτω δεξιά πλευρά του κουμπιού

2021-1-DE02-KA220-ADU-000033587



Κώδικας

Κώδικας MicroPython για το πείραμα:

```
Thonny - Raspberry Pi Pico :: /button.py @ 1:36
                                                                             \times
File Edit View Run Tools Help
 🗋 💕 🛃 🔹 🕸 🐟 🐟 🗈 🖿 🚥
 [button.py]* ×
   1 from machine import Pin #import Pin from library machine
     from time import sleep #import sleep from library time
   4 button = Pin(5, Pin.IN, Pin.PULL_DOWN)
   6 while True:
          if button.value():
   8
              print("Button is pressed!")
              sleep(1)
  10
 Shell >
 MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
 >>> %Run -c $EDITOR CONTENT
  Button is pressed!
```

MicroPython (Raspberry Pi Pico)

Παράδειγμα





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
- 2 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x καλώδιο Micro-USB
- 1 x Βομβητής

Διάγραμμα συνδεσμολογίας



fritzing

συνδέστε το μεγαλύτερο άκρο (+) του βομβητή στον ακροδέκτη
 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
🗋 💕 📓 🔷 🎋 🔿 3. .e 🕨 🧰
[buzzer.py] ×
   1 from machine import Pin #import Pin from library machine
   2 from time import sleep #import sleep from library time
   4 buzzer = Pin(5, Pin.OUT)
   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
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
>>>
                                                              MicroPython (Raspberry Pi Pico)
```

Παράδειγμα





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 \rightarrow Save as..., (Αρχείο \rightarrow Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα pot.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
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x καλώδιο Micro-USB

Διάγραμμα συνδεσμολογίας

1 x Ποτενσιόμετρο



fritzing

Το μαύρο καλώδιο πρέπει να συνδεθεί στον ακροδέκτη GND (Pin 3).

- Το κίτρινο καλώδιο πρέπει να συνδεθεί στον ακροδέκτη 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: The Thonny - Raspberry Pi Pico :: /pot.pv @ 9:1 × File Edit View Run Tools Help 🗋 💕 📓 🔹 🦠 👒 🔍 .et 🕨 📾 [pot.py] × 1 from machine import Pin, ADC #import Pin from library machine from time import sleep #import sleep from library time 4 pot = ADC(Pin(26))6 while True: print(pot.read u16()) sleep(0.5) 9 Shell MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040 Type "help()" for more information. >>> MicroPython (Raspberry Pi Pico)

Παράδειγμα





Προχωρημένα πειράματα

Πριν προχωρήσουμε σε αυτήν την ενότητα και στην επόμενη, πρέπει να κάνουμε μερικές τροποποιήσεις στην πλακέτα διασύνδεσης. Αυτό περιλαμβάνει την προσθήκη μερικών επιπλέον εξαρτημάτων και τη συνδεσιμότητα τους.

Απαιτούμενο υλικό

- Πακέτο μπαταριών 1 x 6-AA
 1 x MB-10
 - 1 x MB-102 μονάδα ισχύος
- 4 x καλώδια jumper αρσενικού σε αρσενικό

Ακολουθήστε το σχηματικό για να συνδέσετε τα νέα στοιχεία:



fritzing

- 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)



Εικόνα Παράδειγμα



5. Μονάδα φωτεινών σηματοδοτών LED

Απαιτούμενο υλικό

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε μια Μονάδα φωτεινών σηματοδοτών LED. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File \rightarrow Save as..., (Αρχείο \rightarrow Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα traffic.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
- 4 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x Μονάδα φωτεινών σηματοδοτών LED
- 1 x καλώδιο Micro-USB



Διάγραμμα συνδεσμολογίας



fritzing

- το πορτοκαλί καλώδιο (R) συνδέεται στην ακίδα GPIO3
- το κίτρινο καλώδιο (Υ) συνδέεται στην ακίδα GPIO4
- το πράσινο καλώδιο (G) συνδέεται στην ακίδα GPIO5

- το μαύρο καλώδιο (GND) συνδέεται στη γραμμή GND ((-) μαύρο)

2021-1-DE02-KA220-ADU-000033587



Κώδικας

Κώδικας MicroPython για το πείραμα:

```
Thonny - Raspberry Pi Pico :: /traffic.py @ 21:1
                                                                  ×
File Edit View Run Tools Help
 🗋 💕 📓 🔹 🎋 👒 🥆 🖈 🖿 🧰
 [traffic.py] ×
     from machine import Pin
     from time import sleep
   4 ledred = Pin(3, Pin.OUT)
   5 ledyellow = Pin(4, Pin.OUT)
   6
      ledgreen = Pin(5, Pin.OUT)
      while True:
   8
          ledred.value(1)
  10
           sleep(5)
          ledyellow.value(1)
          sleep(1)
          ledred.value(0)
          ledyellow.value(0)
          ledgreen.value(1)
          sleep(10)
          ledyellow.value(1)
          ledgreen.value(0)
           sleep(1)
  20
           ledyellow.value(0)
 Shell ×
 MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
 >>>
                                                     MicroPython (Raspberry Pi Pico)
```

Παράδειγμα





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 \rightarrow Save as..., (Αρχείο \rightarrow Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα ldr.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 αντίσταση 2200hm

Διάγραμμα συνδεσμολογίας

- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x φωτοαντίσταση LDR



fritzing

Η αριστερή πλευρά του LDR είναι συνδεδεμένη σε ράγα 3V ((+) κόκκινο)

Η δεξιά πλευρά του LDR συνδέεται με μια αντίσταση 220 Ohm
 και στον ακροδέκτη GPIO26 ADC (κίτρινο καλώδιο)

– η δεξιά πλευρά της αντίστασης συνδέεται με τη ράγα GND ((-) μαύρο)

2021-1-DE02-KA220-ADU-000033587



Κώδικας

Κώδικας MicroPython για το πείραμα:



Παράδειγμα





Κινητήρας συνεχούς ρεύματος (μικρός ανεμιστήρας)

Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε ένα μικρό κινητήρα συνεχούς ρεύματος. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File \rightarrow Save as..., (Αρχείο \rightarrow Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα fan.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 δίοδο

Διάγραμμα συνδεσμολογίας

- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x 7. Κινητήρας συνεχούς ρεύματος (μικρός ανεμιστήρας)
- 1 x TIP-120 τρανζίστορ



2021-1-DE02-KA220-ADU-000033587



Κώδικας

Κώδικας MicroPython για το πείραμα:



Παράδειγμα





8. SG-90 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
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x SG-90 σερβοκινητήρα



fritzing

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

Διάγραμμα συνδεσμολογίας

2021-1-DE02-KA220-ADU-000033587



Κώδικας

Κώδικας MicroPython για το πείραμα:

```
Thonny - Raspberry Pi Pico :: /servo.py @ 13:20
                                                                           X
File Edit View Run Tools Help
🗋 💕 📕 🔷 🌞 😘 3. . e 🕨 🧰
[ servo.py ]
     from machine import Pin, PWM #import Pin and PWM from library machine
   2 from time import sleep #import sleep from library time
   4 pwm = PWM(Pin(28))
   5 pwm.freq(50)
   7 while True:
   8
          for position in range(1000,9000,50):
  9
              pwm.duty_u16(position)
  10
              sleep(0.01)
         for position in range(9000,1000,-50):
              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





9. 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
- 4 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x καλώδιο Micro-USB
- 1 x OLED I2C SSD1306 οθόνη



Διάγραμμα συνδεσμολογίας

fritzing

- το κόκκινο καλώδιο είναι συνδεδεμένο στη γραμμή 3v3 (+)

- το μαύρο καλώδιο είναι συνδεδεμένο στη γραμμή GND (-)
- το πορτοκαλί καλώδιο συνδέεται 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:

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.*

| Isapheny Pi Pice @ COM? Search results microphics sel130 and the full Microby from microphics and 127 Microbythe Microbythem (Search Sol 127 David OLB) digiting Microbythe Microbythem (Search Sol 127 David OLB) digiting | Search on PyPI | × | |
|---|--|---|--|
| Search results microsoftwar usel136 sof136 mag ² /se for MicroBythen microsoftwar usel1327 MicroBythen Maker for S03127 based OLED display | Search on PyPI | - | |
| Search results micropython usef106 sulf156 mig-Teefor MicroPython micropython usef107 MicroPython Mistary for SD01827 based OLED display | | | |
| sad Solid State Drive Checker address: orsubsystem, nel 1368 Crescelly them Reney for SSO 1360 OLEB depitys. Address: OSSI 1360 OLEB depitys. Pythen Reney to use 5200 1360-based 12864 or 128c3 Renetwork for depithem Renc. Stategetters (For explanate Renc. Stategetter). | n 2 pixel OLED displays with a | | |
| | Clear | | |
| | Addriak Sinakathan Milli Constylina Hang for SQUID OLD Adaptop. Adabatati 2020 Kapitop Jan Kapitop Hang Uldek on 12a1 Kapitop Jan Kapitop Hang Uldek on 12a1 Million Makathan Pataster. 100 | <u>aldnah stradgethen and 100</u> Croudlyban Henry for 201303 (LLD deplays). <u>Address 2013</u> Hand Hand Hand Hand Hand Hand Hand Hand | Address instantionen met 132 Construction of the Statistical COLD deploys. Address 2003 Address 2 |



4. Στο επόμενο παράθυρο, κάντε κλικ στο Install.

| lie Edit View Run Tools Help | | | | | | |
|--|---------------------------|---|---|-------|--|--|
| disputiteds (| | | | | | |
| 1 | | | | | | |
| | The Manage packages for R | sipberry Pi Pice @ COM7 | | × | | |
| | esd1306 | | Search on | Py#1 | | |
| | <pre> «RGTALL» </pre> | micropython-ssd1306 Latest table version 0.3 Semmery: un100 mode for MicroPython Anthen Safest Lehrann Manepue Handlighta.com/Information Python and the statest statest statest statest Padage is not available at micropython.org. Vers | napathon-sulf1006 -sulf13002 Seen at PyPI will be installed | | | |
| Shell RicroPython v1.18 on 2022-01-17; Raspb Type "help()" for more information. | | lingall | | Cleve | | |
| MicroPython v1.18 on 2022-01-17; Rasp Type "help()" for more information. | | | | | | |

5. Περιμένετε να εγκατασταθεί το πακέτο και μετά πιέστε 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.pv] ×
   1 from machine import Pin, I2C
   2 import ssd1306
   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)
  14 oled.text("OLED", 0, 0)
  15 oled.text("Display", 0, 10)
  16 oled.text("Hello SmartHome", 0, 30)
  17 oled.text(" People!", 0, 40)
  19 oled.show()
  20
 Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
 >>>
                                                             MicroPython (Raspberry Pi Pico)
```

Παράδειγμα





10. Αναγνώστης RFID RC522

Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αναγνώστη RFID. 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 rfid.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

Διάγραμμα συνδεσμολογίας

- 1 x καλώδιο Micro-USB
- 7 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x RFID RC522 module
- 1 x RFID Κλειδί



fritzing

Το 3ν3 (κόκκινο καλώδιο) είναι συνδεδεμένο στη γραμμή 3ν3
 (+)

- GND (μαύρο καλώδιο) συνδέεται στη γραμμή GND (-)
- RST (κίτρινο καλώδιο) συνδέεται στην ακίδα GPIO0
- SDA (λευκό καλώδιο) συνδέεται στην ακίδα GPIO1.
- SCK (πράσινο καλώδιο) συνδέεται στην ακίδα GPIO2
- Το MOSI (μπλε καλώδιο) συνδέεται στην ακίδα GPIO3.



- MISO (πορτοκαλί καλώδιο) συνδέεται στην ακίδα GPIO4

Κώδικας

Παρόμοια με την οθόνη OLED, χρειαζόμαστε μια πρόσθετη βιβλιοθήκη που ελέγχει τη μονάδα RFID, δηλαδή τη βιβλιοθήκη MFRC522. Μπορούμε να την κατεβάσουμε από το https://github.com/pimylifeup/MFRC522-

python/blob/master/mfrc522/MFRC522.py. Κατεβάστε το αρχείο και ανοίξτε το στο Thonny Python. Έπειτα πατήστε File \rightarrow Save as... επιλέξτε Raspberry Pi Pico και αποθηκεύετε το αρχείο με το όνομα mfrc522.py.

Στη συνέχεια, πρέπει να ταυτοποιήσετε το αναγνωριστικό του κλειδιού (fob) για να λειτουργήσει ο αναγνώστης RFID. Για να το κάνετε αυτό, πρέπει να δημιουργήσετε ένα πρόγραμμα που διαβάζει το RFID Fob και δίνει το αναγνωριστικό του. Ελέγξτε το παρακάτω πρόγραμμα:

```
Thonny - Raspberry Pi Pico :: /fobid.py @ 13:16
                                                                                  Х
File Edit View Run Tools Help
🗋 💕 📓 🔘 🌞 👒 🥆 🕪 🗉
 [rfid.py] × [mfrc522.py] × [fobid.py] ×
   1 from mfrc522 import MFRC522
     import utime
   4 reader = MFRC522(spi id=0,sck=2,miso=4,mosi=3,cs=1,rst=0)
   6 print("Bring RFID FOB closer...")
   7 print("")
  10 while True:
         reader.init()
          (stat, tag_type) = reader.request(reader.REQIDL)
  13
         if stat == reader.OK:
              (stat, uid) = reader.SelectTagSN()
              if stat == reader.OK:
                  fob = int.from_bytes(bytes(uid),"little",False)
                   print("FOB ID: "+str(fob))
                   utime.sleep_ms(500)
 Shell >
 Type "help()" for more information.
 >>> %Run -c $EDITOR CONTENT
  Bring RFID FOB closer ...
  FOB ID: 642819473
                                                               MicroPython (Raspberry Pi Pico)
```



Κάντε κλικ στο Play και σαρώστε το κλειδί. Αυτό θα σας δώσει το αναγνωριστικό του. Τώρα, επιστρέφοντας στο πρόγραμμα rfid.py, θα πρέπει να αλλάξετε τον αριθμό στη γραμμή 20 ώστε να ταιριάζει με το αναγνωριστικό του κλειδιού σας. Στη συνέχεια, κάντε κλικ στο Save και εκτελέστε το πρόγραμμά σας. Κώδικας MicroPython για το πείραμα:



Παράδειγμα





Πειράματα με αισθητήρες

11. 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
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x Αισθητήρας σταγόνας βροχής

Διάγραμμα συνδεσμολογίας



fritzing



Το 3v3V (κόκκινο καλώδιο) είναι συνδεδεμένο στη γραμμή 3v3V
 (+)

- GND (μαύρο καλώδιο) συνδέεται στη γραμμή GND (-)

- DO (πορτοκαλί καλώδιο) συνδέεται στην ακίδα GPIO17

Κώδικας

Κώδικας MicroPython για το πείραμα:





Παράδειγμα





12. 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
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x HC-SR04 Αισθητήρας υπερήχων



fritzing

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 <u>Github</u>. 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
                                                                      X
File Edit View Run Tools Help
🗋 💕 📓 🜔 🎋 🔿 3. .et 🕨 🚥
 [ultrasonic.py] × [hcsr04.py] ×
   1 from hcsr04 import HCSR04
     TRTG = 2
   4 ECHO = 3
   6 sensor = HCSR04(TRIG, ECHO)
   8 distance = sensor.distance cm()
  10 print ("Distance: ", distance, "cm")
 Shell ×
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:

```
Thonny - Raspberry Pi Pico :: /ultrasonic.py @ 21:18
                                                                           П
                                                                                 ×
File Edit View Run Tools Help
🗋 💕 📕 🔹 🎄 👒 🔍 🖈 🖿 🧰
[ultrasonic.pv] ×
   1 from machine import Pin
     import utime
   4 TRIG = Pin(2, Pin.OUT)
   5 ECHO = Pin(3, Pin.IN)
   6 def ultra():
         TRIG.low()
         utime.sleep_us(2)
   8
   9
         TRIG.high()
  10
         utime.sleep_us(5)
         TRIG.low()
         while ECHO.value() == 0:
              signaloff = utime.ticks us()
  14
         while ECHO.value() == 1:
              signalon = utime.ticks us()
          timepassed = signalon - signaloff
         distance = (timepassed * 0.0343) / 2
          print("The distance from object is ",distance,"cm")
  18
  19 while True:
         ultra()
         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





13. 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 \rightarrow Save as..., (Αρχείο \rightarrow 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
- 3 x καλώδια Jumper από αρσενικό σε θηλυκό
- 1 x αισθητήρας κίνησης PIR



fritzing

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


Κώδικας

Κώδικας MicroPython για το πείραμα:



Παράδειγμα





14. Αισθητήρας DHT11

Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα θερμοκρασίας και υγρασίας DHT11. 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 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
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x καλώδιο Micro-USB
- 1 x αισθητήρας θερμοκρασίας DHT11

Διάγραμμα συνδεσμολογίας



fritzing

Το VCC (κόκκινο καλώδιο) είναι συνδεδεμένο στη γραμμή 3ν3
 (+)

- 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 @ 14:1
                                                                  ×
File Edit View Run Tools Help
🗋 💕 📓 🔷 🎄 👒 🛪 ...e 🕨 📾
 [dht11.py] ×
   1 from machine import Pin
   2
     from time import sleep
      import dht
   4
      sensor = dht.DHT11(Pin(5))
      while True:
   8
          sensor.measure()
   9
          temp = sensor.temperature()
          hum = sensor.humidity()
          print("Room temperature: ",temp, "Celcius")
          print("Room humidity: ",hum, "%")
          sleep(5)
  14
 Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
 >>> %Run -c $EDITOR_CONTENT
  Room temperature: 22 Celcius
  Room humidity: 45 %
                                                     MicroPython (Raspberry Pi Pico)
```

Παράδειγμα





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 \rightarrow Save as..., (Αρχείο \rightarrow Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα flame.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

Διάγραμμα συνδεσμολογίας

- 1 x καλώδιο Micro-USB
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x KY-026 αισθητήρας φλόγας



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 @ 11:1
                                                                 X
File Edit View Run Tools Help
🗋 💕 📓 🔹 🕸 🐢 🔍 ...t 🕪  🚥
 [flame.py] ×
   1 from machine import Pin
     from time import sleep
   4 flame = Pin(5, Pin.IN)
   5 sleep(2)
     while True:
   8
         if flame.value() == 0:
   9
             print("Flame Detected")
  10
             sleep(3)
  11
 Shell ×
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
 >>> %Run -c $EDITOR CONTENT
  Flame Detected
  Flame Detected
  Flame Detected
```

MicroPython (Raspberry Pi Pico)

Παράδειγμα





16. MQ-135 Αισθητήρας ανίχνευσης αερίου

Περιγραφή

Σε αυτό το πείραμα θα μάθετε πώς να συνδέετε και να ελέγχετε τον αισθητήρα ανίχνευσης αερίου MQ-135. Ανοίξτε το Thonny Python και μετά μεταβείτε στο File \rightarrow Save as..., (Αρχείο \rightarrow Αποθήκευση ως), επιλέξτε Raspberry Pi Pico και αποθηκεύστε το αρχείο σας με το όνομα gas.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
- 3 x καλώδια jumper αρσενικό σε αρσενικό
- 1 x αισθητήτα ανίχνευσης αερίου MQ-135

Διάγραμμα συνδεσμολογίας



fritzing

Το VCC (κόκκινο καλώδιο) είναι συνδεδεμένο στη γραμμή 3ν3
 (+)

- Το GND (μαύρο καλώδιο) συνδέεται στη γραμμή GND (-)
- DO/OUT (κίτρινο καλώδιο) συνδέεται στην ακίδα GPIO5



Κώδικας

Κώδικας MicroPython για το πείραμα:

```
×
Thonny - Raspberry Pi Pico :: /gas.py @ 11:1
File Edit View Run Tools Help
🗋 💕 🔲 🔷 🎄 🕾 3. . e 🕨 🧰
 [gas.py] ×
   1 from machine import Pin
   2 from time import sleep
   4 gas = Pin(5, Pin.IN)
   5 sleep(2)
   6
   7 while True:
  8
        if gas.value() == 0:
            print("Gas Detected")
   9
  10
             sleep(3)
  11
 Shell >
 MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
 Type "help()" for more information.
 >>> %Run -c $EDITOR CONTENT
  Gas Detected
  Gas Detected
  Gas Detected
```

MicroPython (Raspberry Pi Pico)

Παράδειγμα





ΠΑΡΑΡΤΗΜΑ: Συγκεντρωτικός Πίνακας MicroPython

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

| Ψηφιακή είσοδος | | |
|--|--|--|
| Κλήση της κλάσης ακίδας «Pin class» | from machine import Pin | |
| | <pre>button = Pin(pin_value, Pin.IN)</pre> | pin_value από 0 έως 40 |
| Ενεργοποίηση της ψηφιακής εξόδου του αντικειμένου. | <pre>button = Pin(pin_value), Pin.IN, Pin.PULL_UP)</pre> | Ενεργοποίηση αντίστασης PULL UP |
| | <pre>button = Pin(pin_value), Pin.IN, Pin.PULL_DOWN)</pre> | Ενεργοποίηση αντίστασης PULL DOWN |
| Ανάγνωση Εισόδου | <pre>value = button.value(1)</pre> | 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 ΡWM | from machine import PWM | |
| Ενεονοποίηση της | lod - PWM (Pip (pip Walko) | pin_value από 0 έως 40 |
| αναλογικής εξόδου | frequency) | frequency σε ΗΖ, από 0 έως 78125 |
| Ανάγνωση εισόδου | <pre>led.duty(duty_cycle)</pre> | duty_cycle από 0 έως 1023 (έξοδος ΟV σε έξοδο 3.3V αντίστοιχα) |



| Αναλογική είσοδος | | |
|--|--------------------------------------|---|
| Κλήση της κλάσης ADC | from machine import ADC | |
| Ενεργοποίηση της αναλογικής εισόδου | <pre>pot = ADC(Pin(pin_value))</pre> | pin_value 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) | pot.atten(ADC.ATTN_11DB) | ADC.ATTN_0DB: τάση πλήρους εύρους: 1.2 V |
| | | ADC.ATTN_2_5DB: τάση πλήρους εύρους: 1.5 V |
| | | ADC.ATTN_6DB: τάση πλήρους εύρους: 2.0 V |
| | | ADC.ATTN_11DB: τάση πλήρους εύρους: 3.3 V |
| Δήλωση του εύρους τιμών εισόδου (προεπιλογή 12 bit) | pot.width(ADC.WIDTH_10BIT) | ADC.WIDTH_9BIT : εύρος 0 έως 511 |
| | | ADC.WIDTH_10BIT : εύρος 0 έως 1023 |
| | | ADC.WIDTH_11BIT : εύρος 0 έως 2047 |
| | | ADC.WIDTH_12BIT : εύρος 0 έως 4095 |
| Ανάγνωση εισόδου | <pre>value = pot.readl()</pre> | value (η τιμή) είναι ένας ακέραιος αριθμός από το 0 έως το μέγιστο του εύρους που καθορίζεται από την εντολή ADC.WIDTH_#BIT (βλ. προηγούμενο) |

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



| Δομή Εντολής «If» | | |
|--|---|--|
| if <exprl>: <statement1></statement1></exprl> | <expr#>: η συνθήκη ελέγχου που πρέπει να επιστρέψει True (Σωστό) ή False (Λάθος)</expr#> | |
| <pre>elif <expr2>: <statement2> elif <expr3>:</expr3></statement2></expr2></pre> | <statement#>: σύνολο εντολών που πρέπει να εκτελεστούν όταν ικανοποιείται η σχετική συνθήκη</statement#> | |
| <statement3> ()</statement3> | <expr#> (e.g. το σύνολο <statement2> εκτελείται όταν πληρείται η <expr2>)</expr2></statement2></expr#> | |
| else: <statementn></statementn> | <statementn>: σύνολο εντολών που εκτελούνται όταν δεν πληρείται καμία από τις συνθήκες <expr#></expr#></statementn> | |

| Δομή της εντολής «While loop» | | |
|---|---|--|
| <pre>while <expr>: <statement(s)></statement(s)></expr></pre> | <expr>: η συνθήκη ελέγχου που πρέπει να επιστρέψει True (Σωστό) ή False (Λάθος)</expr> | |
| | <statement#>: σύνολο εντολών που θα εκτελούνται εφόσον πληρείται η συνθήκη <expr></expr></statement#> | |

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



| Διάφορα | | | |
|---------|---|---|--|
| DHT11 | import dht | Εισαγωγή της βιβλιοθήκης DHT | |
| | <pre>sensor = dht.DHT11(Pin(pin_number))</pre> | Ενεργοποίηση της μεταβλητής αισθητήρα με τον αντίστοιχο αριθμό ακίδας pin_number. | |
| | sensor.measure() | Ενημέρωση των μεταβλητών του αισθητήρα | |
| | <pre>temp = sensor.temperature()</pre> | Αποθήκευση τρέχουσας τιμής θερμοκρασίας | |
| | <pre>hum = sensor.humidity()</pre> | Αποθήκευση τρέχουσας τιμής υγρασίας | |
| | from machine import I2C | Εισαγωγή της βιβλιοθήκης Ι2C | |
| | import ssd1306 | Εισαγωγή της βιβλιοθήκης ssd1306 | |
| | i2c = I2C(-1, scl=Pin(1), sda=Pin(0)) | Ενεργοποίηση της μεταβλητής i2c στις ακίδες SCL & SDA του Pico | |
| | oled_width = 128 | | |
| | oled_height = 64 | Ενεργοποίηση της οθόνης | |
| | <pre>oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)</pre> | | |
| Οθόνη | oled.text('Hello, World 1!', 0, 0) | | |
| | oled.text('Hello, World 2!', 0, 10) | Αποθήκευση μηνυμάτων στο buffer οθόνης | |
| | oled.text('Hello, World 3!', 0, 20) | | |
| | oled.show() | Εμφάνιση μηνυμάτων (απαραίτητο για την εμφάνιση των μηνυμάτων που είναι αποθηκευμένα στο buffer οθόνης) | |
| | display.pixel(3, 4, 1) | Θέστε το pixel που βρίσκεται στη θέση (x,y) της οθόνης, με x=3 & y=4, στην κατάσταση 1 (δηλ. εμφάνιση) | |



Co-funded by the European Union





2021-1-DE02-KA220-ADU-000033587

Με τη χρηματοδότηση της Ευρωπαϊκής Ένωσης. Οι απόψεις και οι γνώμες που διατυπώνονται εκφράζουν αποκλειστικά τις απόψεις των συντακτών και δεν αντιπροσωπεύουν κατ'ανάγκη τις απόψεις της Ευρωπαϊκής Ένωσης ή του Ευρωπαϊκού Εκτελεστικού Οργανισμού Εκπαίδευσης και Πολιτισμού (EACEA). Η Ευρωπαϊκή Ένωση και ο EACEA δεν μπορούν να θεωρηθούν υπεύθυνοι για τις εκφραζόμενες απόψεις.

Raspberry Pi Pico Pinout







w: sh4seniors.erasmusplus.website e: info@sh4seniors.erasmusplus.website