



Raspberry Pi Project

ΜΑΘΗΜΑΤΑ RASPBERRY

ΜΑΡΙΟΣ ΒΑΣΙΛΕΙΟΥ

ΔΟΥΜΑ ΑΝΑΣΤΑΣΙΑ

ΚΑΒΑΛΛΙΕΡΑΤΟΥ ΕΡΓΙΝΑ

Εισαγωγή & Στόχοι Μαθήματος

2/34

- ▶ Σε αυτό το μάθημα θα δείτε έτοιμα και εύκολα projects που υπάρχουν στο διαδίκτυο
- ▶ Στόχος είναι η εξοικείωση του χρήστη με τον προγραμματισμό κινητήρων και αισθητήρων

Περιεχόμενα

- ▶ Κατασκευή μικρού οχήματος
- ▶ Κατασκευή ρομπότ που ακολουθεί μια γραμμή

Κατασκευή μικρού οχήματος

Αναλυτικές οδηγίες για το project αυτό μπορείτε να βρείτε στην παρακάτω σελίδα στα αγγλικά:

<https://projects.raspberrypi.org/en/projects/build-a-buggy>

Τι εξαρτήματα θα χρειαστείτε

5/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

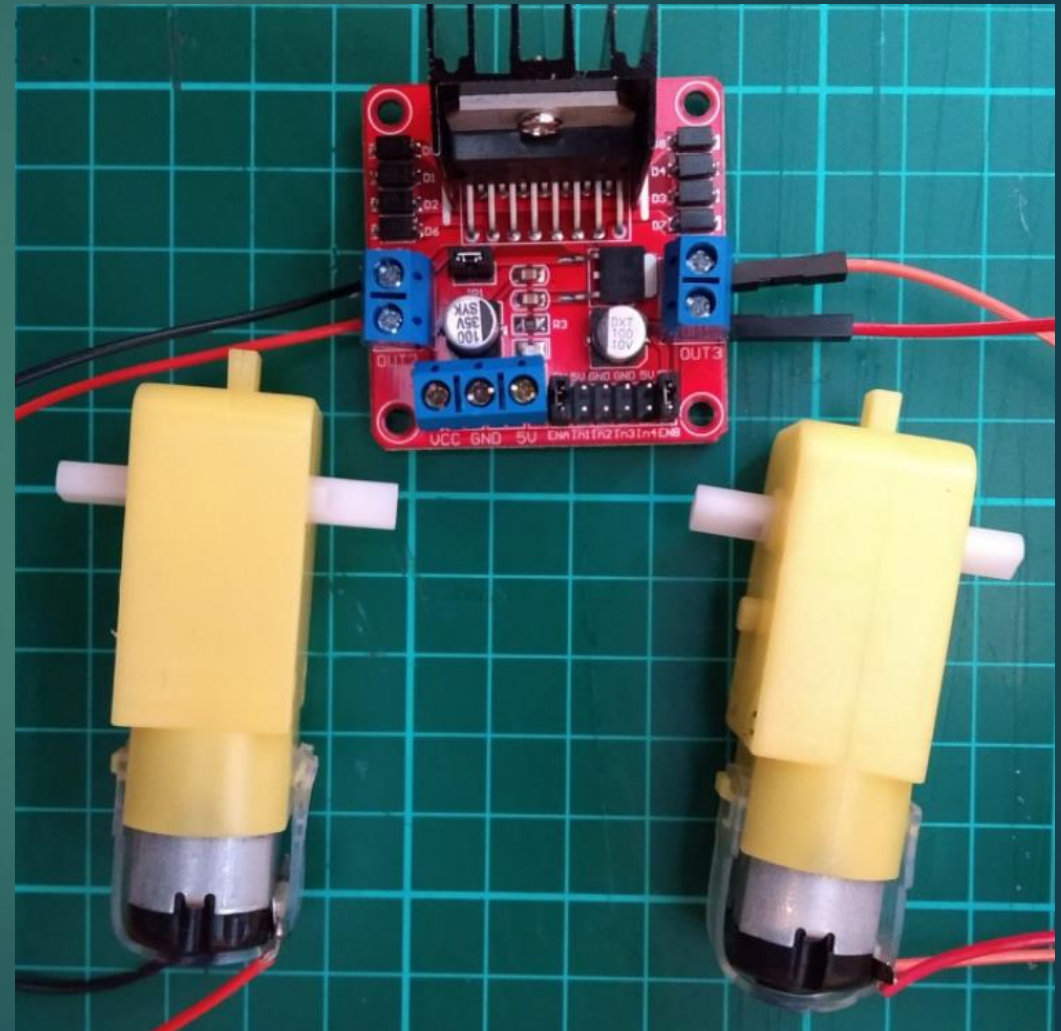
- ▶ Raspberry Pi 3
- ▶ Ελεγκτής Κινητήρων
- ▶ 2 × 3V - 6V DC κινητήρες
- ▶ 2 × Ρόδες
- ▶ 4 × AA μπαταρίες και 1 × θήκη μπαταριών ή Power Bank (2,5+Ampere)
- ▶ Ρόδα τύπου Ball caster
- ▶ Καλώδια μικρά με ακροδέκτες
- ▶ Κάποιο μικρό πλαστικό κουτί που θα λειτουργήσει ως σκελετός

Σύνδεση Κινητήρων

6/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

- ▶ Για τη σύνδεση των κινητήρων θα χρειαστεί να κολλήσετε 2 καλώδια στους ακροδέκτες του κάθε κινητήρα (1 σε κάθε ακροδέκτη)
- ▶ Στην συνέχεια αρκεί να συνδέσετε τις άκρες στις οπές του ελεγκτή κινητήρων με επιγραφή OUT1, OUT2, OUT3, και OUT4 όπως φαίνεται στην φωτογραφία

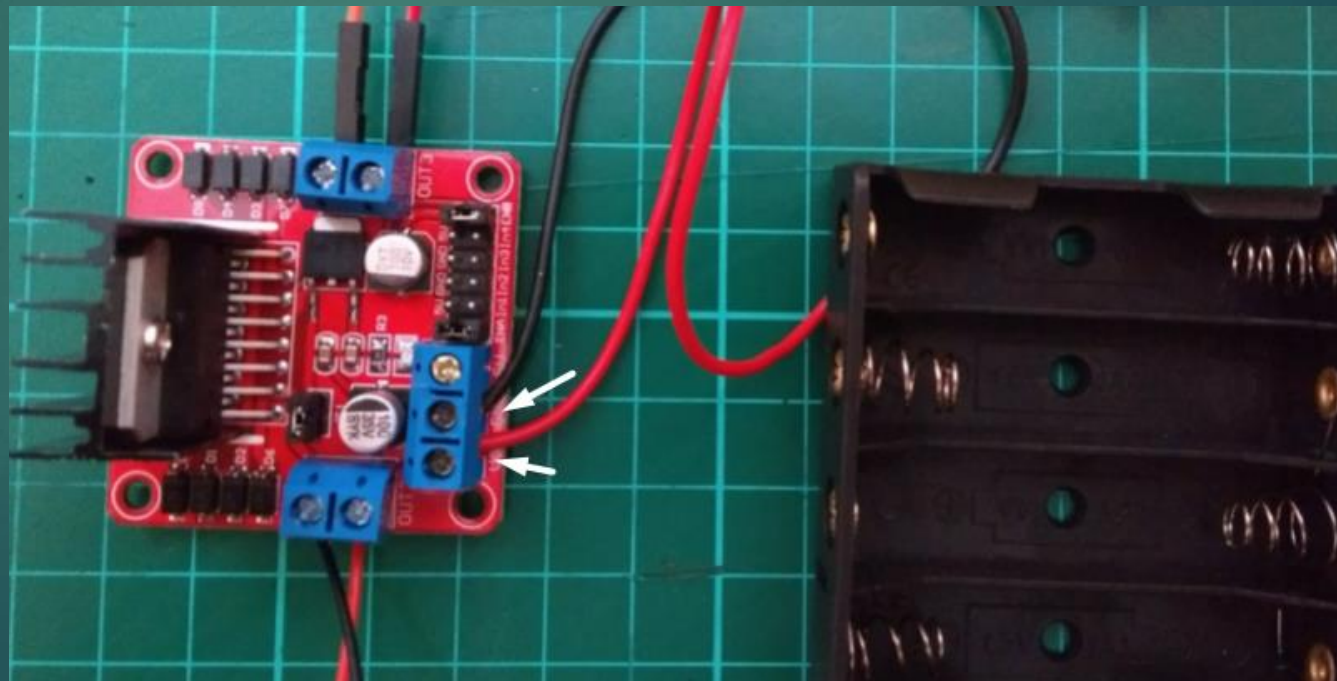


Σύνδεση κινητήρων με μπαταρία

7/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

- ▶ Σε αυτό το σημείο θα πρέπει να συνδέσετε τις άκρες στις οπές του ελεγκτή κινητήρων με επιγραφή VCC, GND, και 5V όπως φαίνεται στην φωτογραφία. Το κόκκινο καλώδιο της μπαταριοθήκης συνδέεται στο VCC ενώ το μαύρο στο GND.

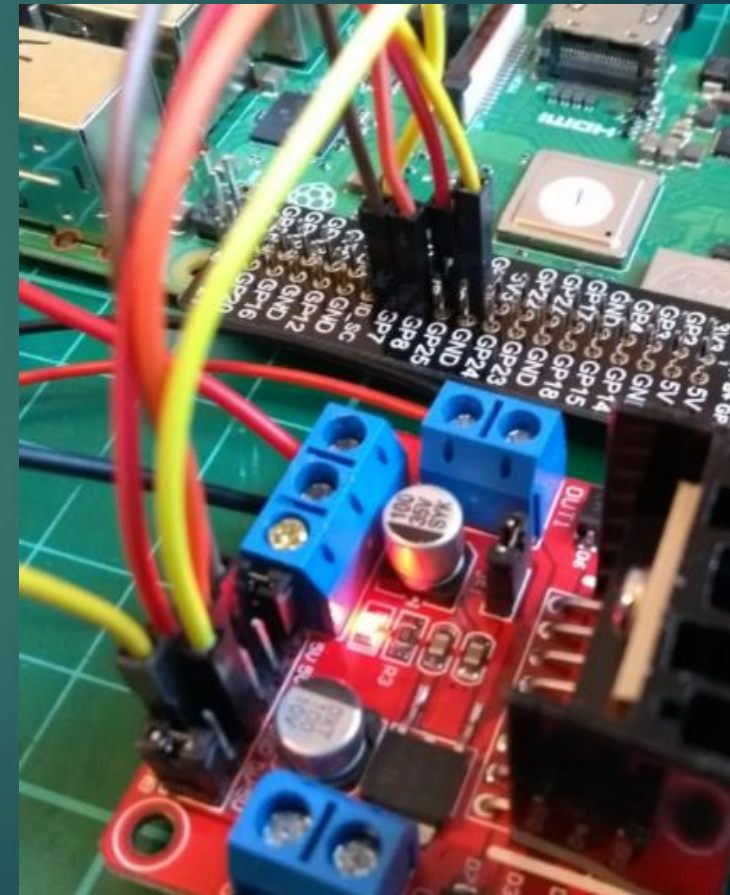


Σύνδεση του ελεγκτή κινητήρων με το Raspberry Pi

8/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

- ▶ Η πλακέτα που χρησιμοποιείται πρέπει να συνδεθεί στο Raspberry Pi. Η κάθε πλακέτα μπορεί να συνδεθεί διαφορετικά για αυτό το λόγο βεβαιωθείτε για την σύνδεση από το εγχειρίδιο που δίνεται από τον κατασκευαστή
- ▶ Στην πλακέτα αυτή υπάρχουν ακροδέκτες με επιγραφές In1, In2, In3, In4 και GND. Σε αυτό το παράδειγμα οι παραπάνω ακροδέκτες θα συνδεθούν με τους ακροδέκτες του Raspberry Pi GPIO 7, 8, 9, 10 και GND αντίστοιχα



Κίνηση

9/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

- ▶ Σε αυτό το παράδειγμα χρησιμοποιείται η κλάση Robot της gpiozero όπως φαίνεται παρακάτω και ορίζονται τα gpio pins:

```
from gpiozero import Robot
robby = Robot(left=(7,8), right=(9,10))
```

- ▶ Με την κλήση των παρακάτω μεθόδων γίνεται εφικτή η κίνηση του ρομπότ:
 - ▶ Κίνηση μπροστά: `.forward()`
 - ▶ Κίνηση πίσω: `.backwards()`
 - ▶ Κίνηση δεξιά: `.right()`
 - ▶ Κίνηση αριστερά: `.left()`
 - ▶ Σταματάει: `.stop()`
- ▶ Κατά την κλήση των παραπάνω μεθόδων μπορεί να οριστεί και η ταχύτητα προσθέτοντας έναν δεκαδικό αριθμό μεταξύ 0 και 1. (πχ. 70% = 0.7)

Κίνηση – Παρατηρήσεις

10/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

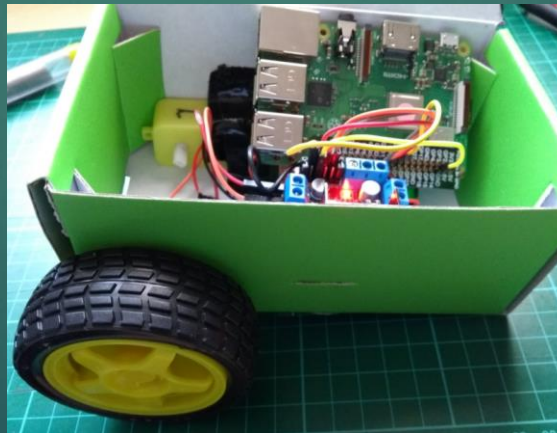
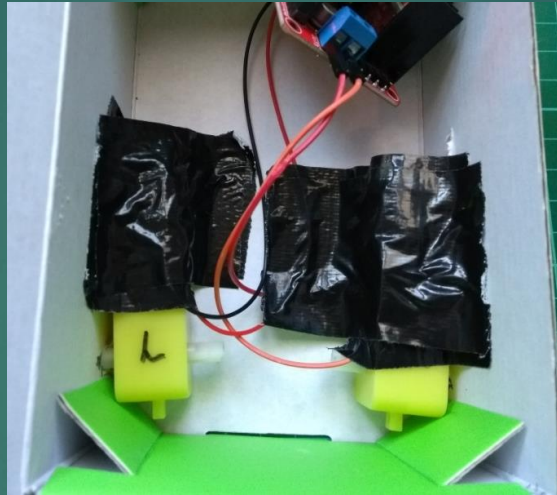
- ▶ Εάν κάποιος από τους κινητήρες κινείται ανάποδα από την εντολή τότε αρκεί να αντιστρέψετε το ζεύγος των gpio pins
- ▶ Π.χ.: Αρχικά ορίσαμε "robby = Robot(left=(7,8), right=(9,10))" , ας υποθέσουμε ότι ο αριστερός κινητήρας κινείται αντίστροφα από το κανονικό (όταν δίνουμε εντολή forward κινείται προς τα πίσω). Για να λύσουμε αυτό το πρόβλημα αντιστρέφουμε τις τιμές στην παρένθεση στο left και θα έχουμε: "robby = Robot(left=(8,7), right=(9,10))"

Κατασκευή αμαξιδίου

11/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

- ▶ Για την κατασκευή του σκελετού του αμαξιδίου ο μόνος περιορισμός είναι η φαντασία σας
- ▶ Σε αυτό το project χρησιμοποιήθηκε ένα μικρό χαρτόκουτο και μονωτική ταινία όπως φαίνεται και στις φωτογραφίες



Άσκηση – Προγραμματισμός ρομπότ

12/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

- ▶ Να δημιουργήσετε πρόγραμμα στο οποίο το αμαξάκι να εκτελεί συνέχεια τις παρακάτω κινήσεις (μέσα σε επανάληψη):
 - Κίνηση προς τα εμπρός για 3 δευτερόλεπτα
 - Κίνηση προς τα δεξιά για 1 δευτερόλεπτο
- ▶ Για τον συνδυασμό των κινήσεων θα χρειαστεί να χρησιμοποιήσετε την εντολή `sleep()` της βιβλιοθήκης `time` και στην παρένθεση προσθέτετε τον αριθμό σε δευτερόλεπτα

Λύση

13/34

Project: <https://projects.raspberrypi.org/en/projects/build-a-buggy>

```
from gpiozero import Robot
from time import sleep
robot = Robot(left = (7, 8), right = (9, 10))
while True:
    robot.forward()
    sleep(3)
    robot.stop()
    robot.right()
    sleep(1)
    robot.stop()
```

Όπως φαίνεται στον κώδικα χρησιμοποιείται η επανάληψη *while* και επαναληπτικά

(i) ενεργοποιεί τους κινητήρες για κίνηση προς τα εμπρός

(ii) δεν εκτελεί κάτι για 3 δευτερόλεπτα

(iii) απενεργοποιεί τους κινητήρες

(iv) ενεργοποιεί τους κινητήρες για κίνηση προς τα δεξιά

(v) δεν εκτελεί κάτι για 1 δευτερόλεπτο

(vi) σταματάει και ξεκινάει από την αρχή

Κατασκευή ρομπότ που ακολουθεί μια γραμμή

Αναλυτικές οδηγίες για το project αυτό μπορείτε να βρείτε στην παρακάτω σελίδα στα αγγλικά:

<https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

Τι εξαρτήματα θα χρειαστείτε

15/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Σε συνέχεια του αμαξιδίου που υλοποιήσατε θα χρειαστείτε τα εξής υλικά:
 - 2 αισθητήρες υπέρυθρων τύπου line following
 - Καλώδια μικρά με ακροδέκτες

Σύνδεση Αισθητήρων

16/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Κάθε αισθητήρας έχει τρεις ακροδέκτες: VCC και GND για το ρεύμα και DO και την έξοδο του σήματος
- ▶ Συνδέστε το VCC στον ακροδέκτη 5V του Raspberry Pi
- ▶ Συνδέστε το GND στον ακροδέκτη GND του Raspberry Pi
- ▶ Συνδέστε το DO του πρώτου αισθητήρα στον ακροδέκτη GPIO 17 του Raspberry Pi
- ▶ Συνδέστε το DO του δεύτερου αισθητήρα στον ακροδέκτη GPIO 27 του Raspberry Pi

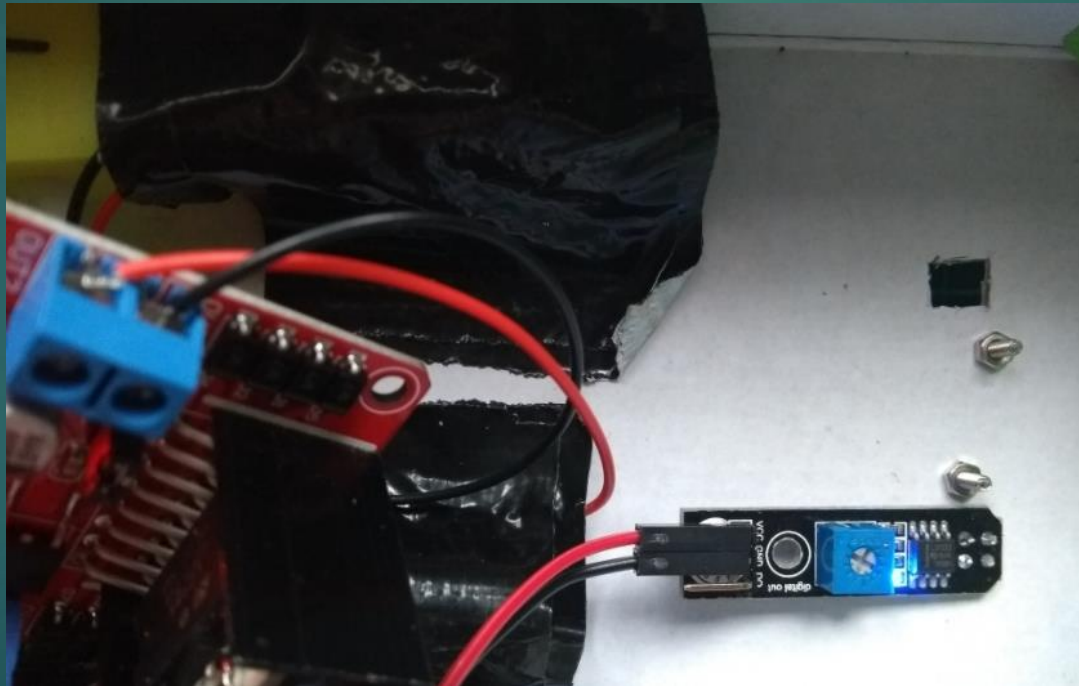


Προετοιμασία σκελετού

17/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Θα πρέπει να ανοίξετε 2 μικρές τρύπες στο κάτω μέρος του κουτιού για να τοποθετηθούν οι αισθητήρες.

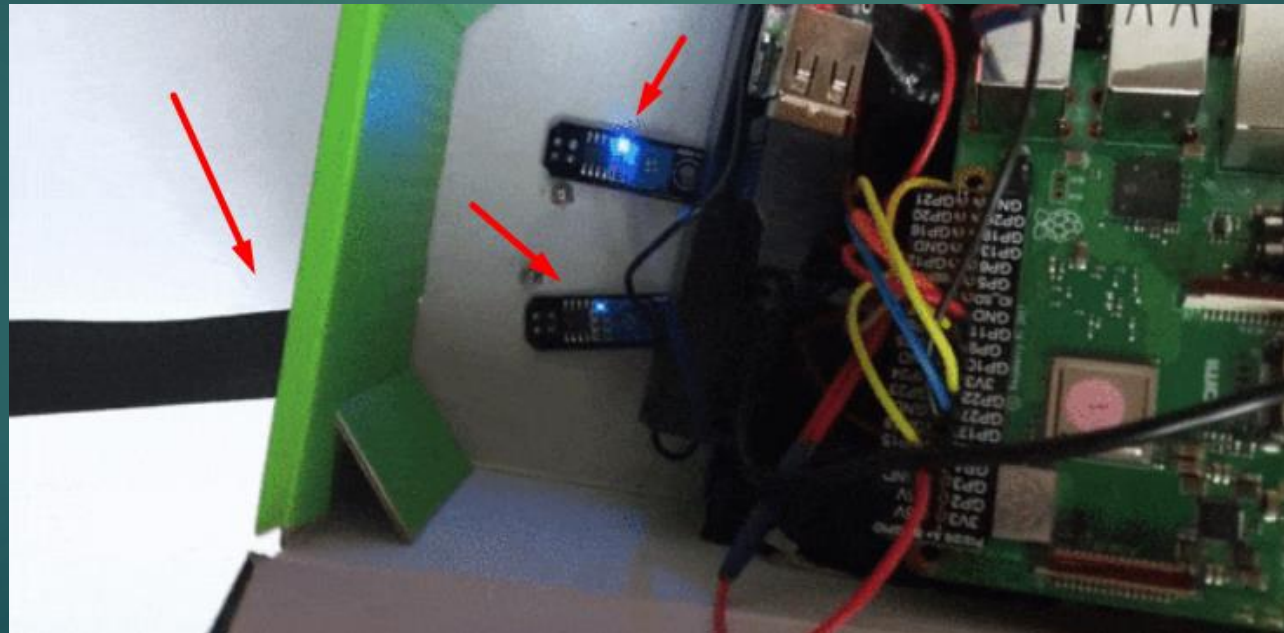


Δοκιμή

18/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Αφού τους τοποθετήσετε, δοκιμάστε να τους περάσετε πάνω από μια μαύρη γραμμή. Όταν ο αισθητήρας περνάει πάνω από την μαύρη γραμμή θα πρέπει να σβήνει το λαμπάκι όπως φαίνεται παρακάτω:

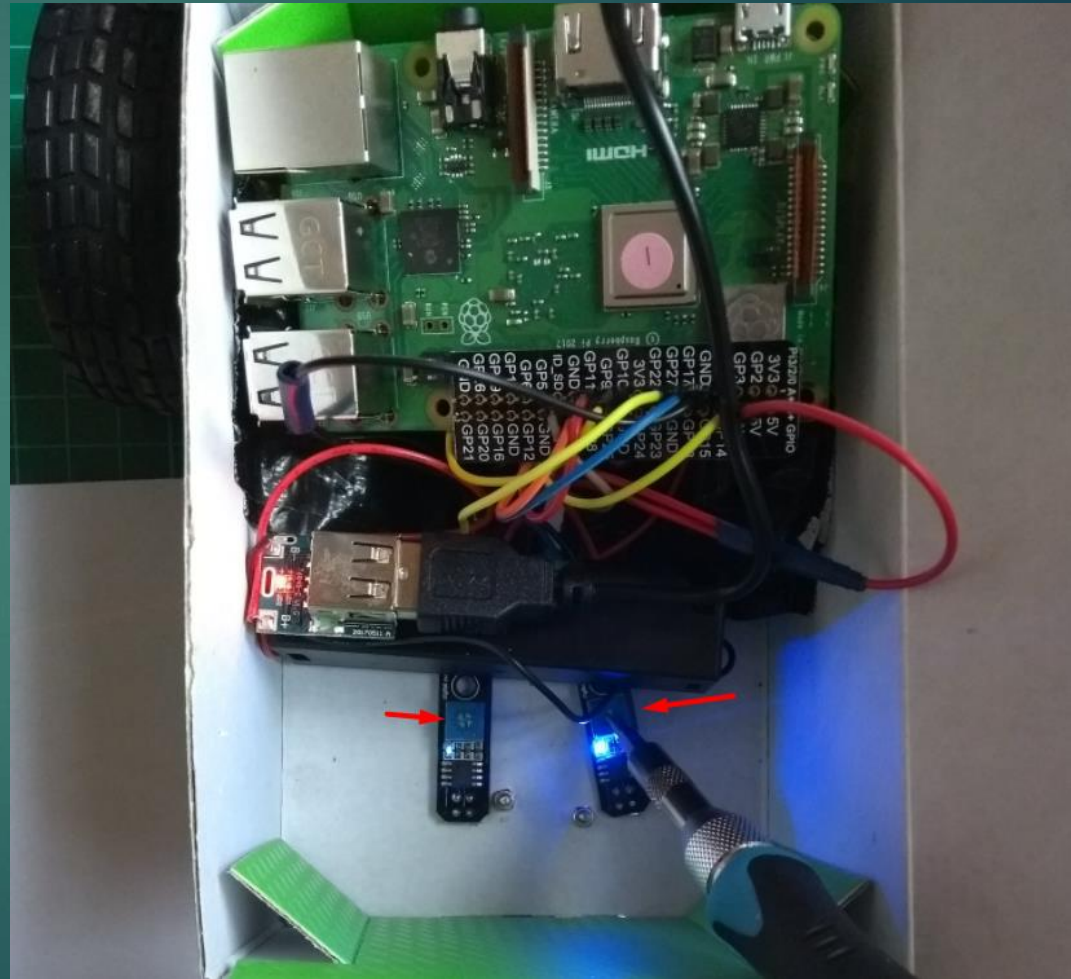


Δοκιμή και παραμετροποίηση

19/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Εάν δεν σβήνει το λαμπάκι θα πρέπει να στρίψετε το ποτενσιόμετρο μέχρι να σβήσει.



Προγραμματισμός (1/2)

20/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Ανοίξτε το “mu” από το *Raspberry Pi Programming menu* και ξεκινήστε να ρυθμίζετε τον ελεγκτή κινητήρων και τους αισθητήρες με την χρήση της βιβλιοθήκης *gpiozero*

```
from gpiozero import Robot, LineSensor
from signal import pause
from time import sleep

robot = Robot(left=(7, 8), right=(9, 10))
left_sensor = LineSensor(17)
right_sensor = LineSensor(27)
```

- ▶ Σε αυτό το παράδειγμα ο ελεγκτής κινητήρων έχει συνδεθεί με τέτοιο τρόπο ώστε ο αριστερός κινητήρας να αντιστοιχεί στα GPIO pins 7 και 8, ενώ ο δεξιός κινητήρας στα GPIO pins 9 και 10
- ▶ Οι αισθητήρες έχουν συνδεθεί: Ο αριστερός στο GPIO pin 27, ενώ ο δεξιάς στο GPIO pin 27

Προγραμματισμός (2/2)

21/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Για αρχή, κατά τον έλεγχο ότι λειτουργεί μπορεί να χρησιμοποιηθεί πολύ απλός κώδικας. Η βιβλιοθήκη `gpiozero` μπορεί να καλέσει κάποια συνάρτηση όταν ο αισθητήρας εντοπίζει την γραμμή:

```
left_sensor.when_line = function_name_to_call  
left_sensor.when_no_line = other_function_name_to_call
```

- ▶ Με το “`left_sensor.when_line = όνομα_συνάρτησης`” θα καλέσει την συνάρτηση που θα ορίσουμε όταν εντοπίζει γραμμή, αντίστοιχα όταν δεν εντοπίζει γραμμή μπορεί να καλέσει κάποια άλλη συνάρτηση
- ▶ Π.χ. αν θέλουμε να ορίσουμε όταν δεν εντοπίζει ο αριστερός αισθητήρας γραμμή να κινείται ευθεία αρκεί ο κώδικας:

```
left_sensor.when_line = robot.forward()
```

Άσκηση

22/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Με βάση τα προηγούμενα να γράψετε κώδικα έτσι ώστε το ρομπότ να εκτελεί τα παρακάτω:
 - Αν ο αριστερός αισθητήρας εντοπίζει γραμμή, να στρίβει αριστερά
 - Αν ο δεξιός αισθητήρας εντοπίζει γραμμή, να στρίβει δεξιά
 - Αν δεν εντοπίζει γραμμή σε κανέναν αισθητήρα να κινείται ευθεία

Λύση

23/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Επειδή ο κώδικας είναι πολύ απλός υπάρχει πιθανότητα να βγει εκτός πορείας. Αυτό που πρέπει να κάνετε είναι απλά να το τοποθετήσετε πάλι πάνω στην γραμμή
- ▶ Θα χρειαστεί να προσθέσετε την `pause()` για να μην τερματίσει το πρόγραμμα

```
left_sensor.when_line = robot.left
right_sensor.when_line = robot.right
left_sensor.when_no_line = robot.forward
right_sensor.when_no_line = robot.forward

pause()
```

Προγραμματισμός – Κατανόηση Κίνησης (1/3)

24/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Η προηγούμενη άσκηση ήταν μια πολύ απλή εφαρμογή line following και παρακάτω θα δούμε πως μπορούμε να την βελτιώσουμε
- ▶ Όταν ο αισθητήρας είναι πάνω από μια γραμμή, δίνει ως έξοδο την τιμή 1, ενώ όταν δεν έχει γραμμή, την τιμή 0
- ▶ Οι ελεγκτές κινητήρων λειτουργούν διαφορετικά, όταν λαμβάνουν την τιμή 1 τότε κινούνται προς τα εμπρός, ενώ όταν λαμβάνουν την τιμή -1 κινούνται αντίστροφα

Προγραμματισμός – Κατανόηση Κίνησης (2/3)

25/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

Παρακάτω θα δούμε τις καταστάσεις που βρίσκεται το ρομπότ:

- ▶ Το ρομπότ βρίσκεται ακριβώς πάνω στην γραμμή και πρέπει να κινείται ευθεία:
 - Και οι 2 αισθητήρες δεν εντοπίζουν την γραμμή και δίνουν ως έξοδο 0
 - Και οι 2 κινητήρες θα έχουν ως είσοδο 1 για να κινηθούν ευθεία
- ▶ Το ρομπότ έχει στρίψει προς τα αριστερά και πρέπει να το επαναφέρουμε με κίνηση προς τα δεξιά:
 - Ο δεξιός αισθητήρας εντοπίζει την γραμμή και δίνει ως έξοδο 1
 - Ο αριστερός αισθητήρας δεν εντοπίζει την γραμμή και δίνει ως έξοδο 0

Προγραμματισμός – Κατανόηση Κίνησης (3/3)

26/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Ο αριστερός κινητήρας πρέπει να κινηθεί προς τα πίσω δίνοντας του την τιμή -1 ή πρέπει να σταματήσει δίνοντάς του τιμή 0
- ▶ Ο δεξιός κινητήρας πρέπει να κινηθεί προς τα εμπρός δίνοντας του την τιμή 1
- ▶ Το ρομπότ έχει στρίψει προς τα δεξιά και πρέπει να το επαναφέρουμε με κίνηση προς τα αριστερά:
 - ▶ Ο δεξιός αισθητήρας δεν εντοπίζει την γραμμή και δίνει ως έξοδο 0
 - ▶ Ο αριστερός αισθητήρας εντοπίζει την γραμμή και δίνει ως έξοδο 1
 - ▶ Ο αριστερός κινητήρας πρέπει να κινηθεί προς τα εμπρός δίνοντας του την τιμή 1
 - ▶ Ο δεξιός κινητήρας πρέπει να κινηθεί προς τα πίσω δίνοντας του την τιμή -1 ή πρέπει να σταματήσει δίνοντάς του τιμή 0

Προγραμματισμός - Δοκιμές

27/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Ο παρακάτω κώδικας διαβάζει τις τιμές από τους αισθητήρες και τις εμφανίζει στην οθόνη
- ▶ Για να επιτευχθεί αυτό χρησιμοποιείται ένας ατέρμονας βρόχος (while True)
- ▶ Για να δείτε να αλλάζουν τιμές μετακινήστε το ρομπότ δεξιά – αριστερά

```
from gpiozero import Robot, LineSensor
from time import sleep

robot = Robot(left=(7, 8), right=(9, 10))
left_sensor = LineSensor(17)
right_sensor = LineSensor(27)

while True:
    left_detect = int(left_sensor.value)
    right_detect = int(right_sensor.value)
    print(left_detect, right_detect)
```

Προγραμματισμός - Προετοιμασία

28/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Μέσα στον ατέρμονα βρόχο θα πρέπει να δημιουργήσετε δύο μεταβλητές, την *left_mot* και την *right_mot*. Αυτοί οι κινητήρες θα αποθηκεύουν την κίνηση του κάθε κινητήρα 1 για κίνηση προς τα εμπρός, 0 για να σταματάει και -1 για κίνηση προς τα πίσω:
 - Αν και οι 2 αισθητήρες έχουν έξοδο 0, τότε και οι 2 κινητήρες πρέπει να λάβουν την τιμή 1
 - Αν ο δεξιός αισθητήρας έχει έξοδο 1, τότε ο αριστερός κινητήρας θα πρέπει να λάβει την τιμή -1
 - Αν ο αριστερός αισθητήρας έχει έξοδο 1, τότε ο δεξιός κινητήρας θα πρέπει να λάβει την τιμή -1

Άσκηση

29/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Με βάση τις προηγούμενες διαφάνειες να φτιάξετε πρόγραμμα το οποίο επαναληπτικά να λαμβάνει δεδομένα από τους αισθητήρες και να εμφανίζει στην οθόνη τις κατάλληλες εντολές (διαφ. 28) έτσι ώστε να ακολουθεί την γραμμή.

Λύση

30/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

```
while True:
    left_detect = int(left_sensor.value)
    right_detect = int(right_sensor.value)

    if left_detect == 0 and right_detect == 0:
        left_mot = 1
        right_mot = 1

    if left_detect == 0 and right_detect == 1:
        left_mot = -1
    if left_detect == 1 and right_detect == 0:
        right_mot = -1

    print(right_mot, left_mot)
```

Μέσα σε ατέρμονα βρόχο:

Διαβάζει τις τιμές των αισθητήρων

Αν και οι 2 αισθητήρες δεν εντοπίζουν γραμμή ($==0$) τότε κινούνται και οι 2 κινητήρες προς τα εμπρός.

Αν ο δεξιός αισθητήρας έχει έξοδο 1 και ο αριστερός 0, τότε ο αριστερός κινητήρας θα πρέπει να λάβει την τιμή -1.

Αν ο αριστερός αισθητήρας έχει έξοδο 1 και ο δεξιός 0, τότε ο δεξιός κινητήρας θα πρέπει να λάβει την τιμή -1.

Προγραμματισμός – Κίνηση (1/3)

31/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

Αφού έχουμε τις τιμές που θα παίρνουν οι κινητήρες μπορούμε να τις περάσουμε στους κινητήρες.

Στο προηγούμενο πρόγραμμα κάνουμε τις εξής αλλαγές:

- ▶ Τον κώδικα το βάζουμε μέσα σε μια συνάρτηση (`def motor_speed()`)
- ▶ Με την εντολή `yield(right_mot, left_mot)` δίνει εντολή να κινηθούν οι κινητήρες

```
def motor_speed():
    while True:
        left_detect = int(left_sensor.value)
        right_detect = int(right_sensor.value)
        ## Stage 1
        if left_detect == 0 and right_detect == 0:
            left_mot = 1
            right_mot = 1
        ## Stage 2
        if left_detect == 0 and right_detect == 1:
            left_mot = -1
        if left_detect == 1 and right_detect == 0:
            right_mot = -1
        #print(r, l)
        yield (right_mot, left_mot)
```

Προγραμματισμός – Κίνηση (2/3)

32/34

Project: <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

- ▶ Το μόνο που μένει είναι να εκτελεστεί η εντολή:

```
robot.source = motor_speed()
```

- ▶ Για να μην εκτελείται ο κώδικας συνέχεια θα προσθέσετε τις παρακάτω εντολές:
 - ▶ `sleep(60)` **#μετά από 60 δευτερόλεπτα θα τρέξει τα παρακάτω**
 - ▶ `robot.stop()` **#σταματάει το ρομπότ**
 - ▶ `robot.source = None` **# δεν θα εκτελεί την συνάρτηση**
 - ▶ `robot.close()`
 - ▶ `left_sensor.close()`
 - ▶ `right_sensor.close()`

Προγραμματισμός – Κίνηση (3/3) Πλήρες πρόγραμμα

- ▶ *Project:*
<https://projects.raspberrypi.org/en/projects/rpi-python-line-following>

33/34

```
from gpiozero import Robot, LineSensor
from time import sleep

robot = Robot(left=(7, 8), right=(9, 10))
left_sensor = LineSensor(17)
right_sensor = LineSensor(27)

speed = 0.65

def motor_speed():
    while True:
        left_detect = int(left_sensor.value)
        right_detect = int(right_sensor.value)
        ## Stage 1
        if left_detect == 0 and right_detect == 0:
            left_mot = 1
            right_mot = 1
        ## Stage 2
        if left_detect == 0 and right_detect == 1:
            left_mot = -1
        if left_detect == 1 and right_detect == 0:
            right_mot = -1
        #print(r, l)
        yield (right_mot * speed, left_mot * speed)

robot.source = motor_speed()

sleep(60)
robot.stop()
robot.source = None
robot.close()
left_sensor.close()
right_sensor.close()
```

Βιβλιογραφία

[1] *Project:* <https://projects.raspberrypi.org/en/projects/build-a-buggy>

[2] *Project:* <https://projects.raspberrypi.org/en/projects/rpi-python-line-following>