

DIY Robot kit for educators

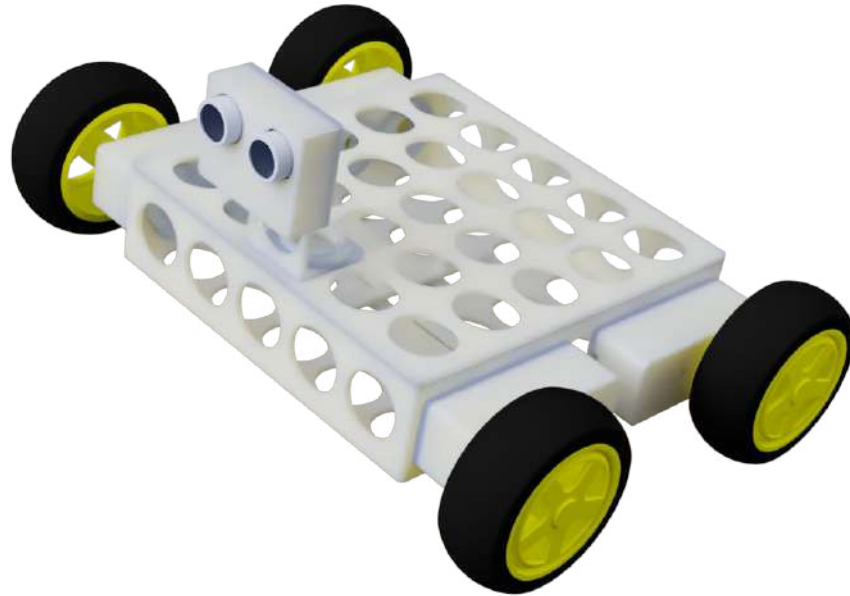
Proteas the modular robot

Google Summer of Code

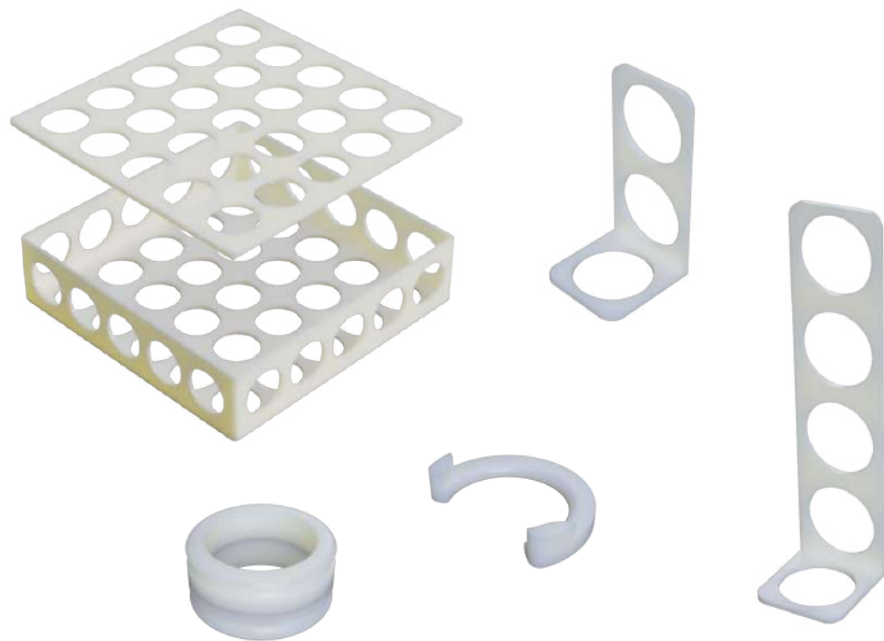


2019

Proteas



Ο σχεδιασμός



Χαρακτηριστικά σχεδιασμού:

- Εύκολη και σταθερή συναρμολόγηση χωρίς βίδες και εργαλεία
- Αρθρωτή σχεδίαση
- Δυνατότητα αναδιάταξης εξαρτημάτων ανάλογα με τις ανάγκες της εφαρμογής.





Mini Breadboard

Ultrasonic Sensor



Dc Motor



Light Sensor



Gyro,
Accelerometer,
Compass Sensor



Servo Motor



IR Obstacle Sensor



Gyro, Accelerometer
Sensor



Raspberry Pi



Arduino



Raspberry Pi
Camera



LCD Screen



Optical Odometer



Ir Receiver



Noise Sensor



Gas Sensor



Buzzer

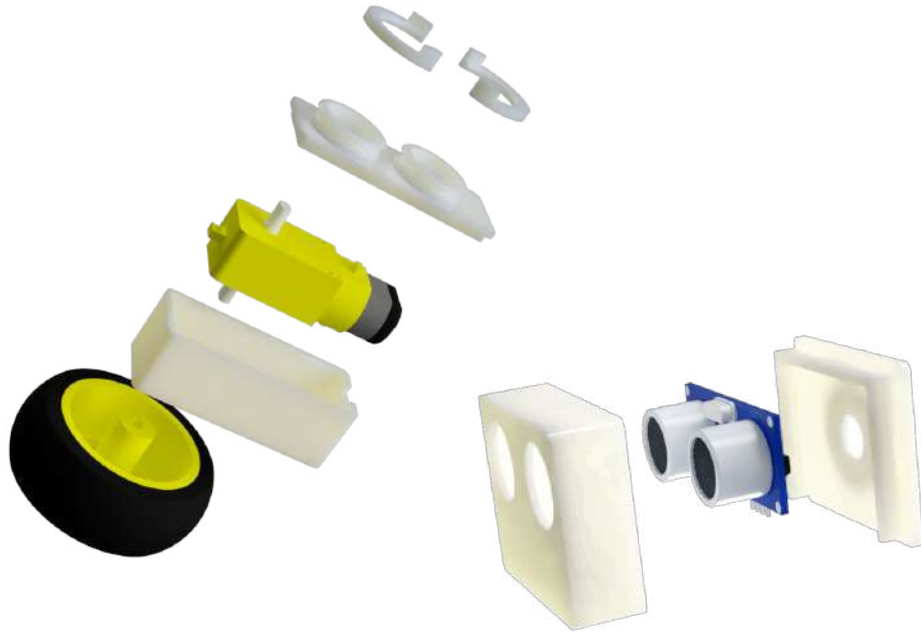


Motion Sensor



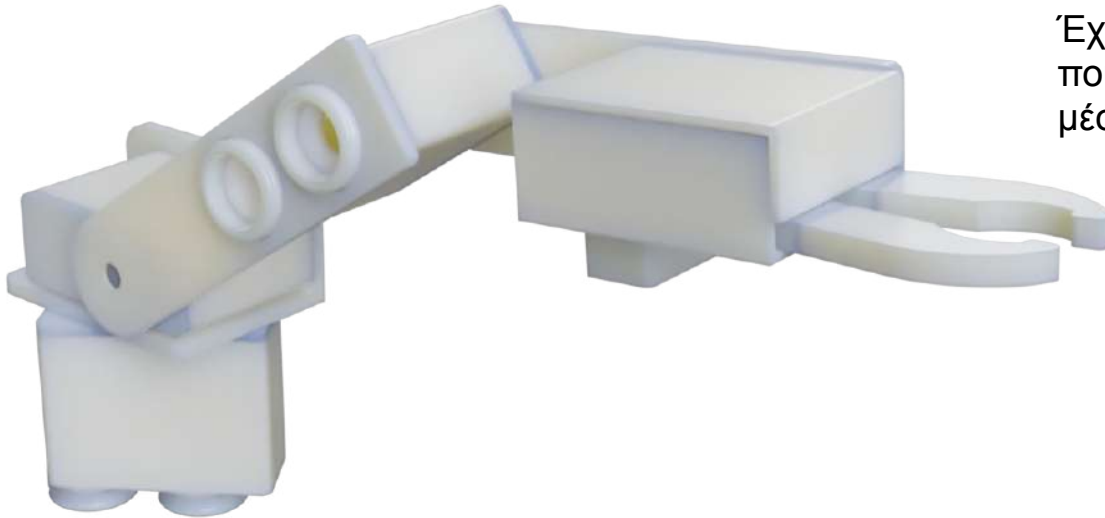
Temperature,
Humidity Sensor

Συναρμολόγηση



- Ειδικές θήκες για τα περισσότερα εξαρτήματα για την εύκολη τοποθέτηση.
- Προστασία των ηλεκτρονικών εξαρτημάτων.
- Γρήγορη συναρμολόγηση.

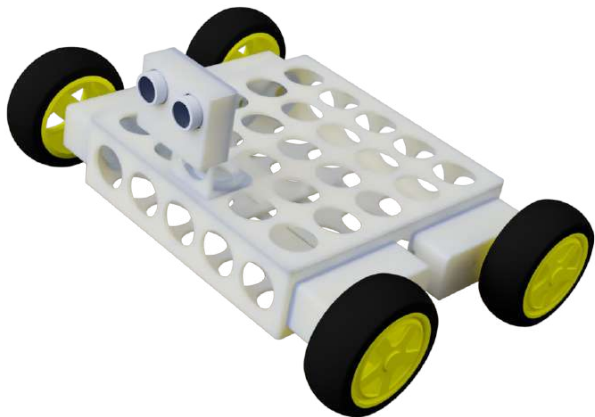
Βραχίονας



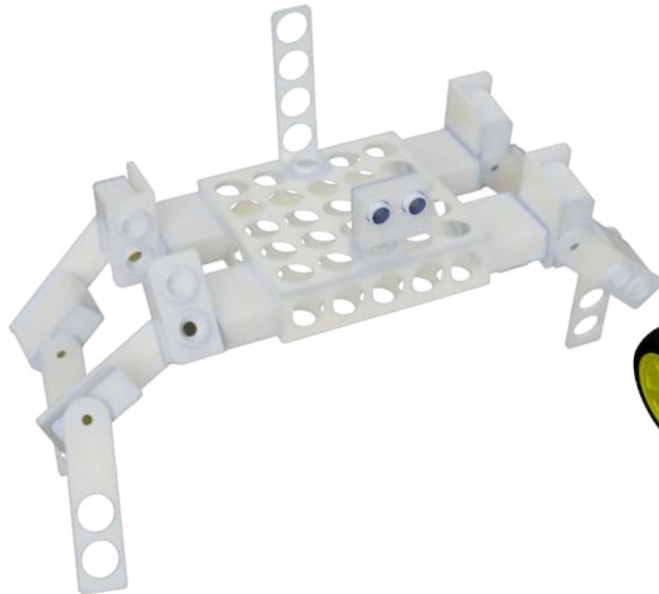
Έχει σχεδιαστεί ρομποτικός βραχίονας που δίνει την δυνατότητα ελέγχου του, μέσω εύκολου προγραμματισμού.



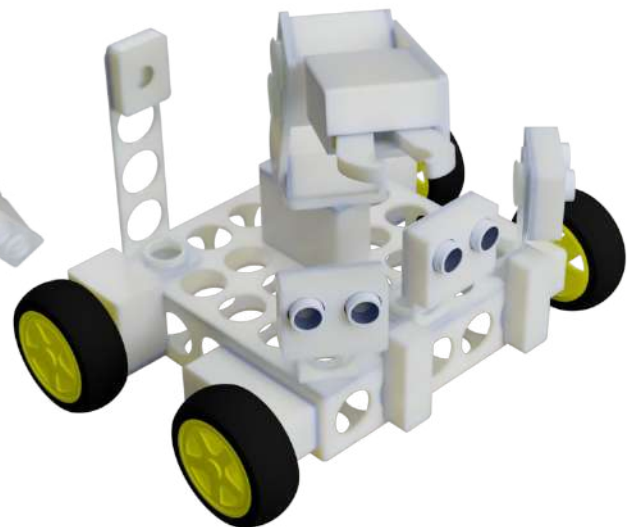
Πιθανές διατάξεις



Διάταξη με τέσσερις
κινητήρες και αισθητήρα
εμποδίων.



Διάταξη σε τετράποδο



Διάταξη με δύο κινητήρες
και πολλαπλούς
αισθητήρες

Η Βιβλιοθήκη

```
class motor():
    """
    The motor Class create new motor objects. For every motor you should create a new object
    e.x. left_motor = motor(pwm pin, pin a , pin b) also you can insert the frequense as freq of pwm
    and duty cicle as dc. The dc parameter controls the speed of the motor and accepts values 0 - 100.
    By default the constructor set the frequense on 1000 Hz and the duty cicle on 50.
    """

    def __init__(self, speed_pin, terma_pin, termb_pin, freq=10000, dc=50):
        GPIO.setup(speed_pin, GPIO.OUT)
        GPIO.setup(terma_pin, GPIO.OUT)
        GPIO.setup(termb_pin, GPIO.OUT)
        self.terma_pin = terma_pin
        self.termb_pin = termb_pin
        self.mot = GPIO.PWM(speed_pin, freq)
        self.freq = freq
        self.dc = dc
        self.dir_control("forward")
        self.mot.start(0)

    def control_speed(self, speed):
        self.mot.ChangeDutyCycle(speed)

    def set_speed(self, speed):
        self.dc = speed

    def dir_control(self, direction):
        if direction == "forward":
            GPIO.output(self.terma_pin, GPIO.HIGH)
            GPIO.output(self.termb_pin, GPIO.LOW)
        elif direction == "reverse":
            GPIO.output(self.terma_pin, GPIO.LOW)
            GPIO.output(self.termb_pin, GPIO.HIGH)
        else:
            print("Motor accepts only forward and reverse values")

    def move(self, direction = "forward"):
        self.dir_control(direction)
        self.mot.ChangeDutyCycle(self.dc)

    def stop(self):
        self.mot.ChangeDutyCycle(0)
```

Βασικές εντολές:

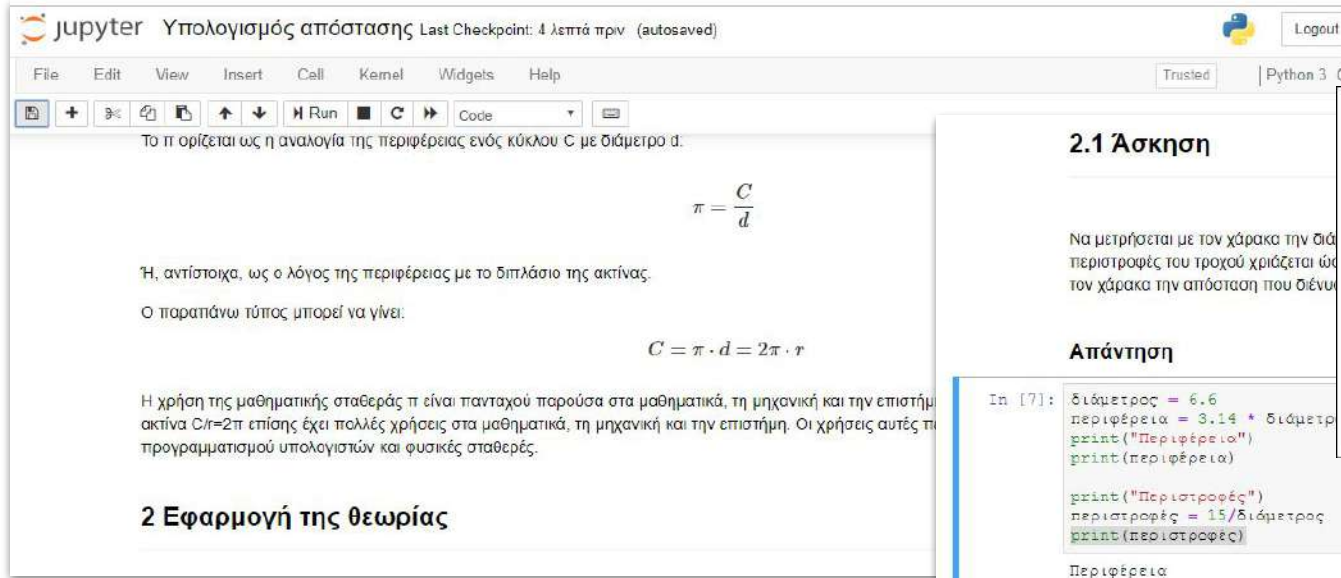
```
motor_a = robot.motor(11,12,13)
motor_a.move()
motor_a.stop()
```

Επιπλέον:

```
motor_a.move("reverse")
motor_a.set_speed(70)
```



Περιβάλλον προγραμματισμού



Το π ορίζεται ως η αναλογία της περιφέρειας ενός κύκλου C με διάμετρο d.

$$\pi = \frac{C}{d}$$

Ή, αντίστοιχα, ως ο λόγος της περιφέρειας με το διπλάσιο της ακτίνας.

Ο παραπάνω τύπος μπορεί να γίνει:

$$C = \pi \cdot d = 2\pi \cdot r$$

Η χρήση της μαθηματικής σταθεράς π είναι πανταχού παρούσα στα μαθηματικά, τη μηχανική και την επιστήμη. Η ακτίνα $C/2\pi$ επίσης έχει πολλές χρήσεις στα μαθηματικά, τη μηχανική και την επιστήμη. Οι χρήσεις αυτές του π προγραμματισμού υπολογιστών και φυσικές σταθερές.

2 Εφαρμογή της θεωρίας

2.1 Άσκηση

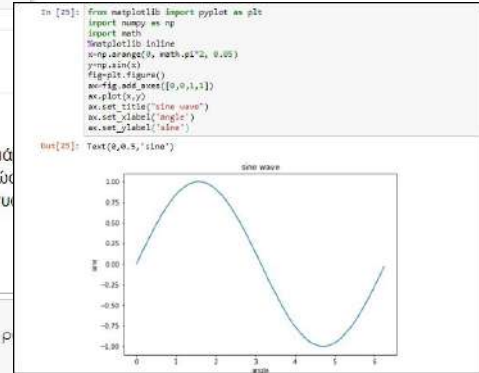
Να μετρήσετε με τον χάρακα την διάμετρο του τροχού χριζείται ως τον χάρακα την απόσταση που διένυσε ο τροχός.

Απάντηση

```
In [7]: διάμετρος = 6.6
περιφέρεια = 3.14 * διάμετρος
print("Περιφέρεια")
print(περιφέρεια)

print("Περιστροφές")
περιστροφές = 15/διάμετρος
print(περιστροφές)

Περιφέρεια
3.14
Περιστροφές
2.28
```



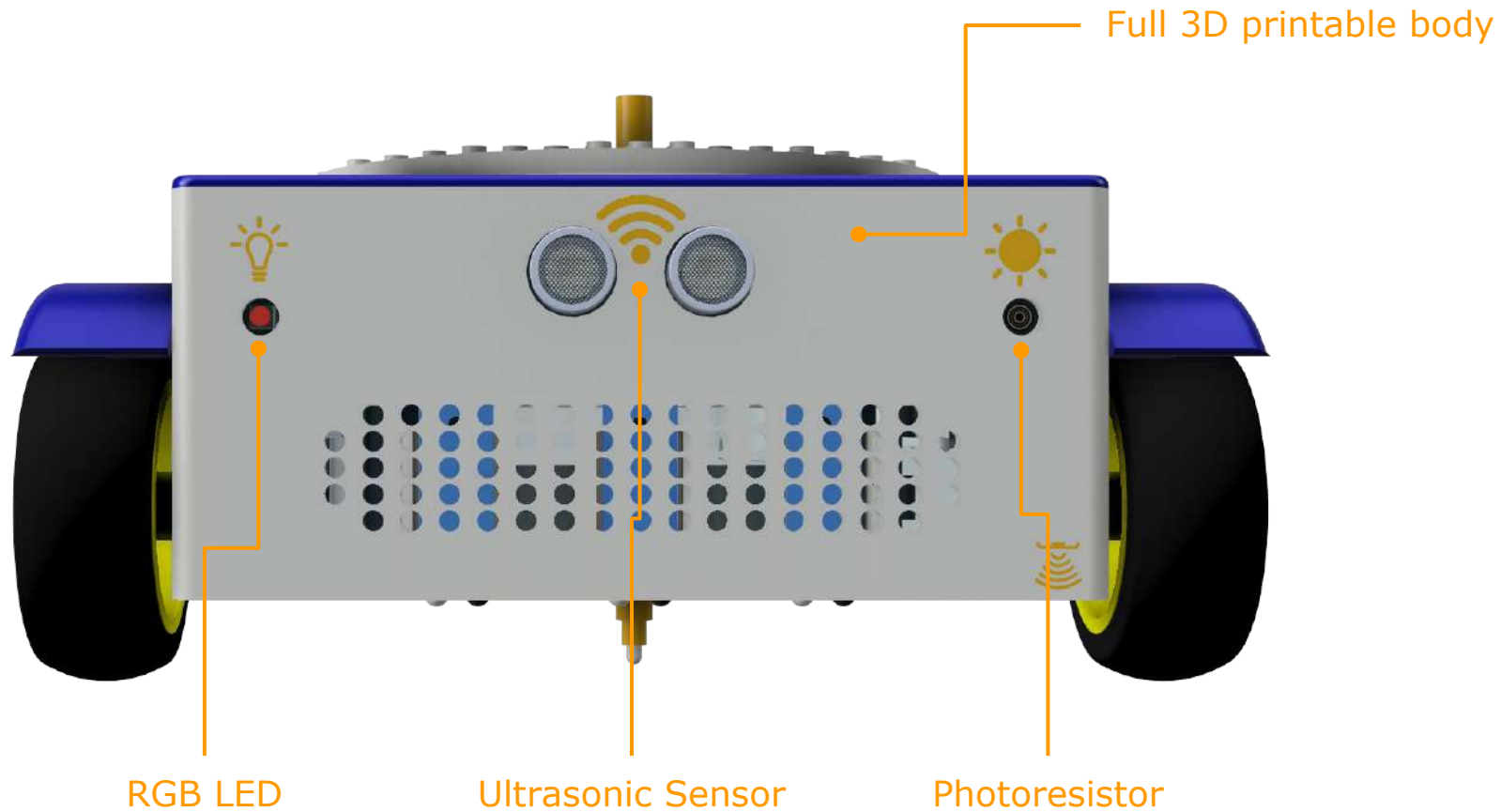
<https://github.com/eellak/gsoc2019-diyrobot>



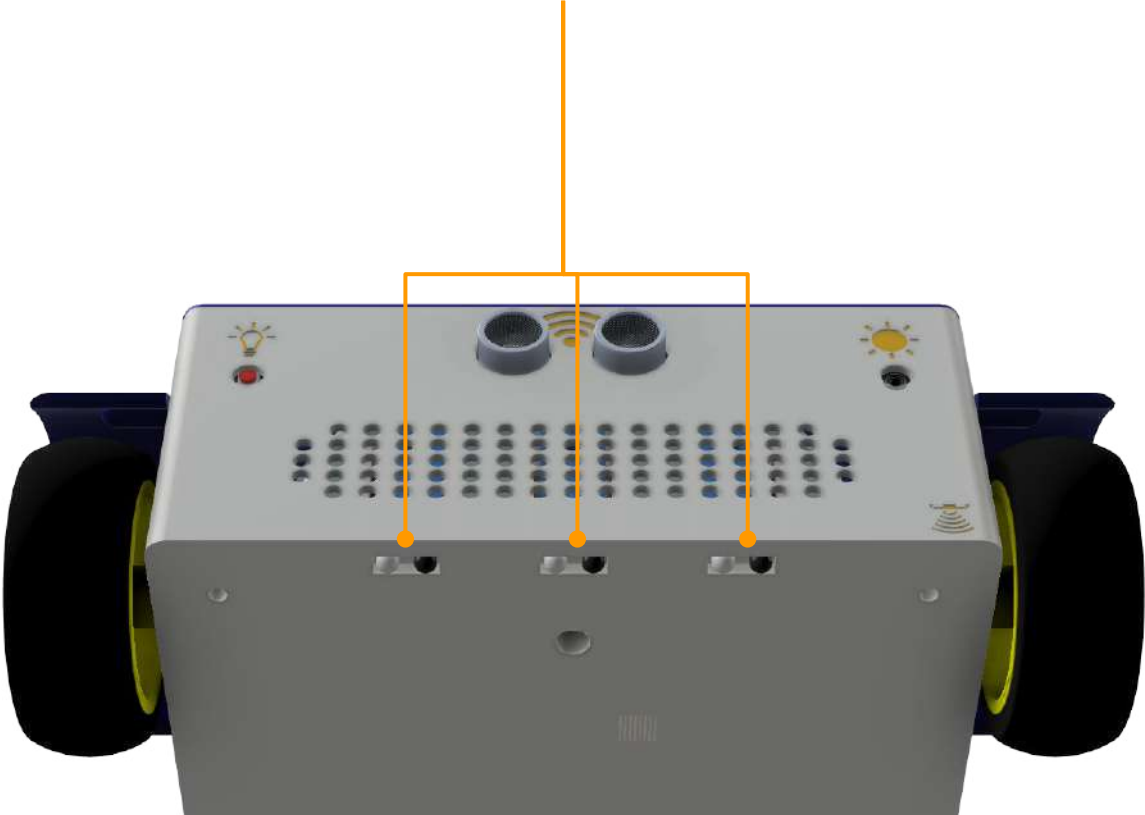


ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

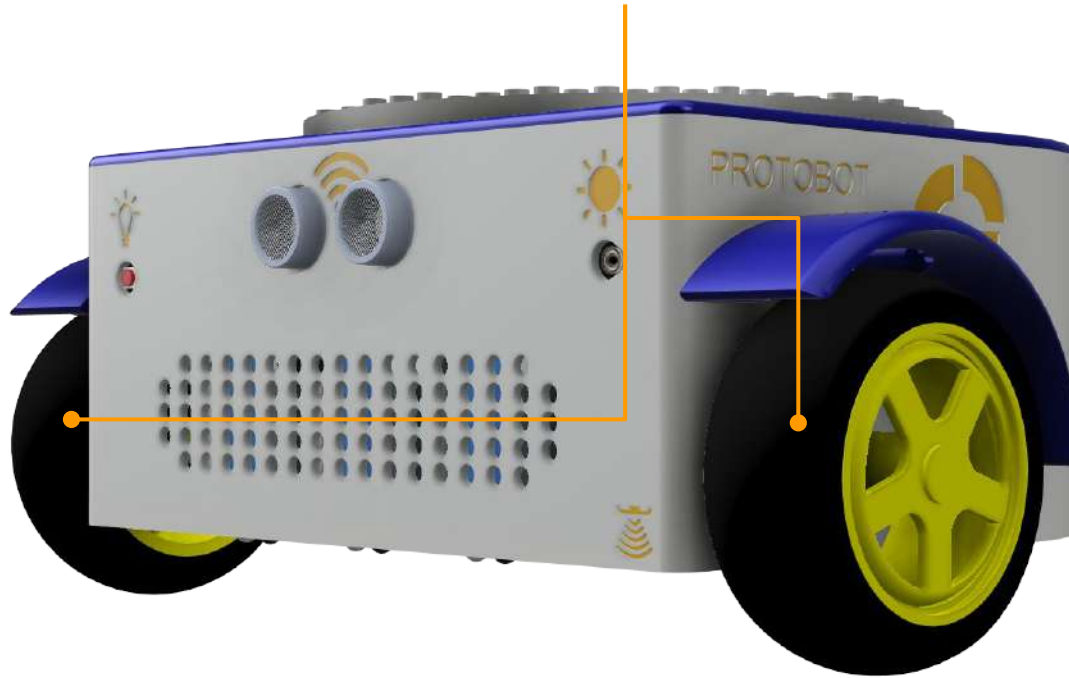
FOSSbot

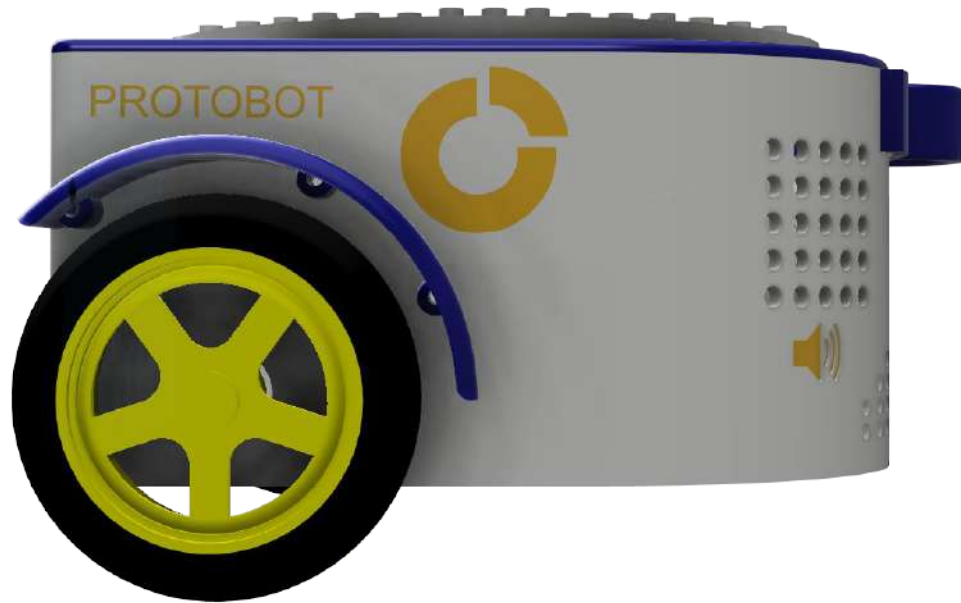


IR Obstacle Sensor (Line following)

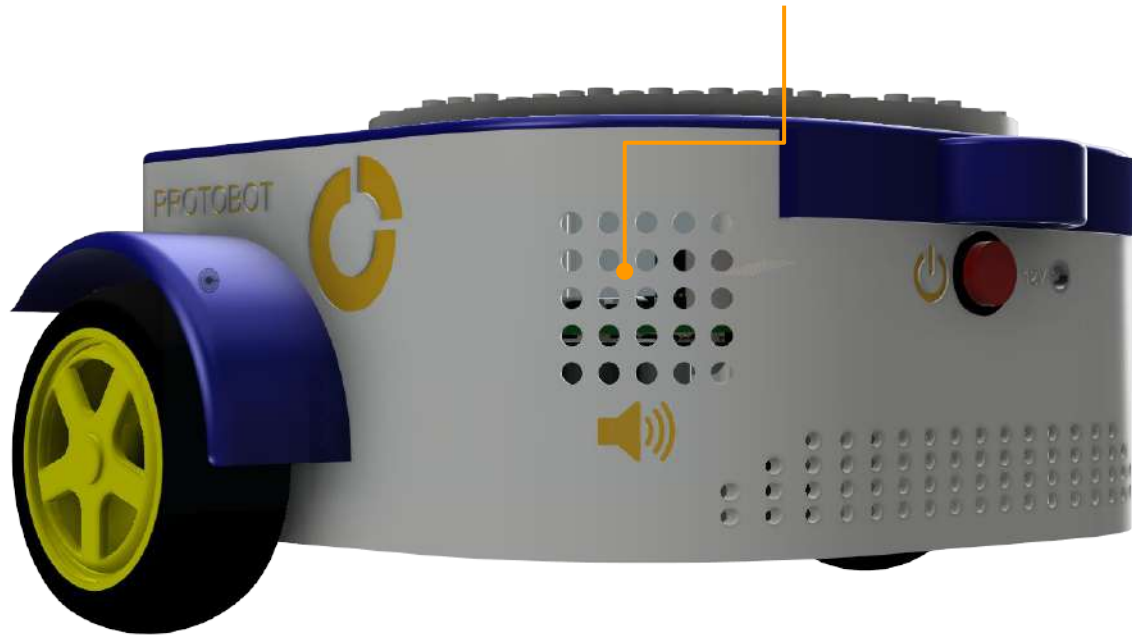


Two motors motion with odometers

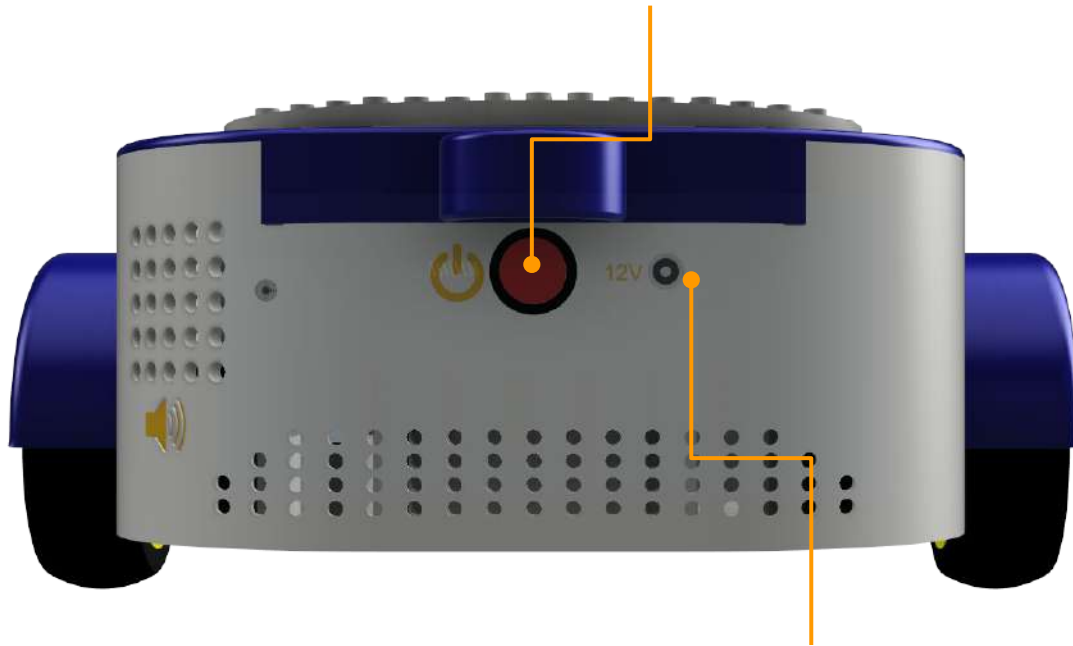




Speaker



On/Off Switch



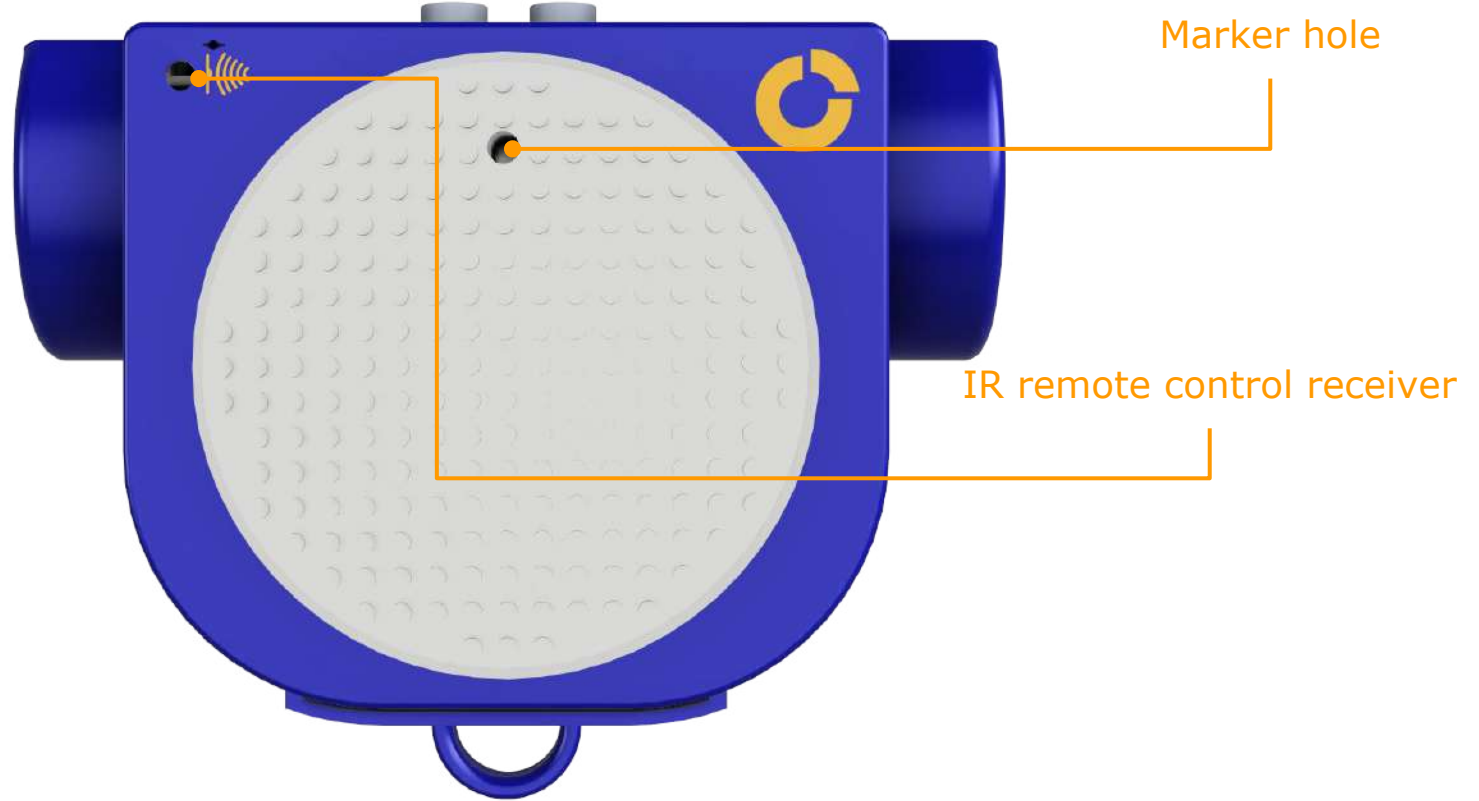
12V Charging port for 3 lite-ion 18650 batteries

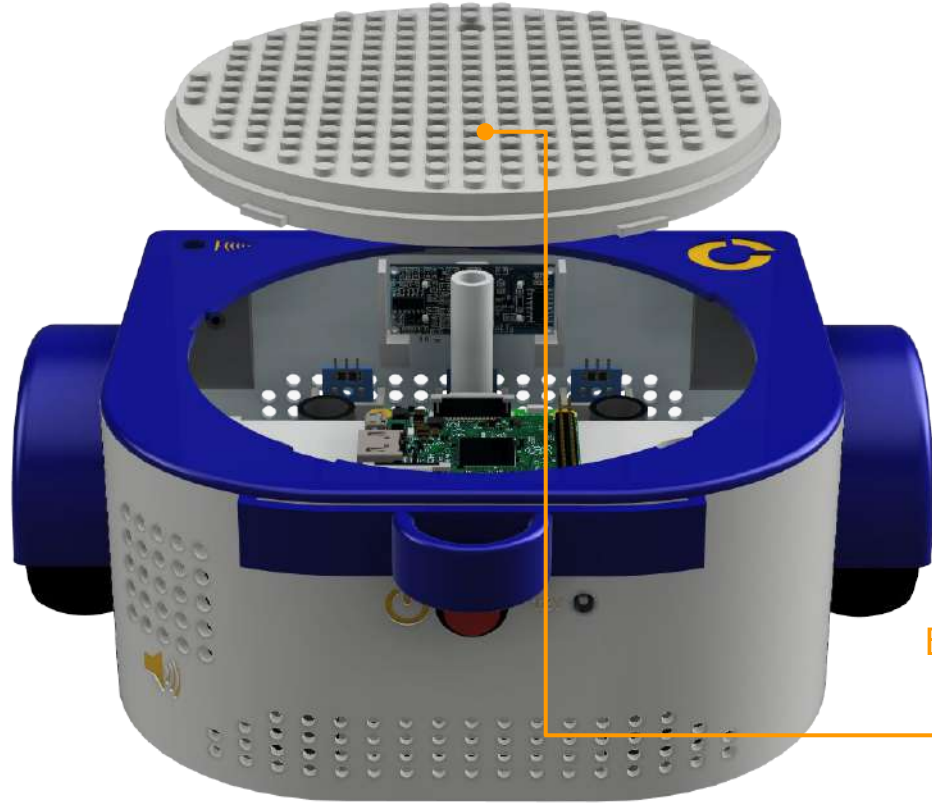




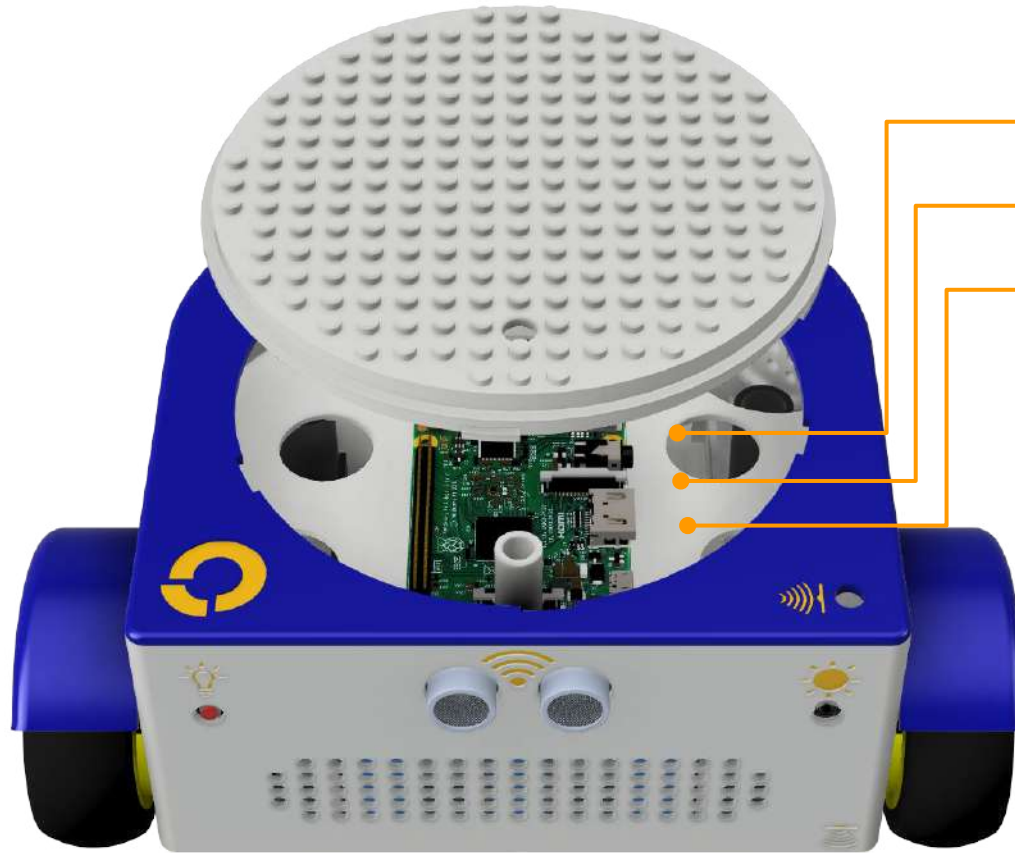
Lego compatible base

Pulling component





Easily removable cover



Accelerometer

Gyroscope

Battery sensor

Sensors:

- Ultrasonic distance sensor
- Battery Sensor
- Accelerometer
- Gyroscope
- Odometers
- IR Receiver
- Line detection sensors
- Light Sensors

Interaction Features:

- Speaker
- Front RGB LED

General Features:

- Lego compatible surface
- Hole in front for marker/ pencil attachment
- Special pulling loop
- Rechargeable batteries



Blockly



- Λογική
- Δομές επανάληψης**
- Συναρτήσεις
- Μαθηματικά
- Κείμενο
- Αισθητήρες

Μέχρι να φτάσεις το [] επανάλαβε []

επανάλαβε [] φορές

κάνε []

επανάλαβε [10] φορές

κάνε []

επανάλαβε ενώ []

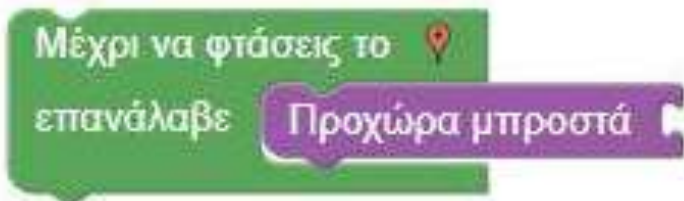
κάνε []

μέτρησε με [i] από το [1] έως το [10] ανά [1]

κάνε []

για κάθε στοιχείο [i] στη λίστα []

Τρέξε το πρόγραμμα!





ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY



Contribute

Harokopio University Developing Team:

- Assistant Professor Iraklis Varlamis
- Phd Candidate Christos Chronis
- Undergraduate Student Athanasios Apostolidis
- Graduate Student Eleftheria Papageorgiou



<https://github.com/eellak/fossbot>