

רובולגו

הקדמה

המעבדה משתמשת ברובוט הבנוי ממערכת LEGO® MINDSTORMS™ Robots, ובשפת התכנות Python, בשימוש בחבילת התוכנה next-python. בחבילה זאת הקוד מתקמפל במחשב הפיתוח, והפקודות לרובולגו נשלחות ממחשב הפיתוח בעזרת תקשורת BlueTooth.

נושאי המעבדה:

תכנות ושימושי חיישנים במערכת RoboLego.

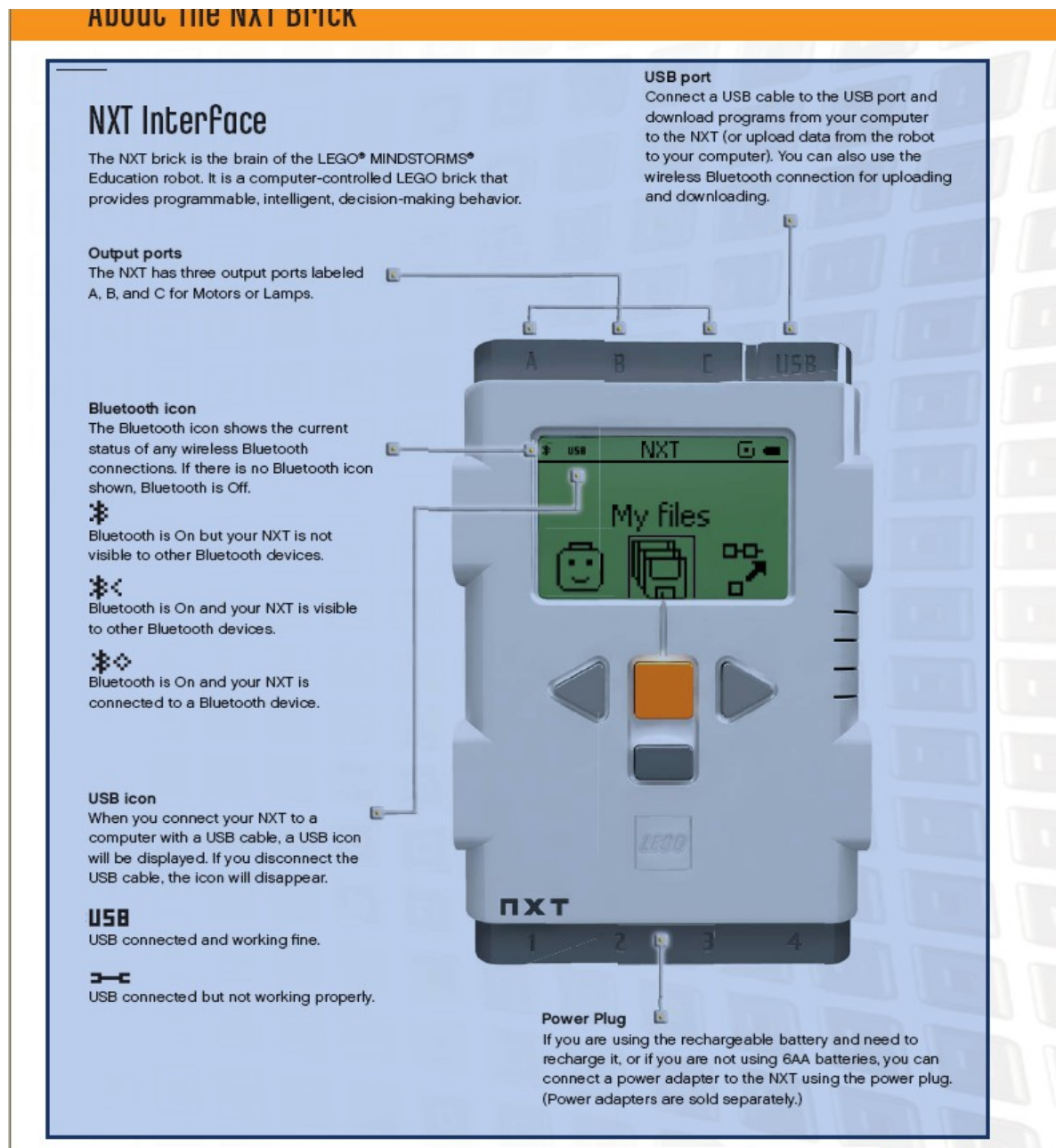
מטרות המעבדה:

תכנות אפליקציות שונות

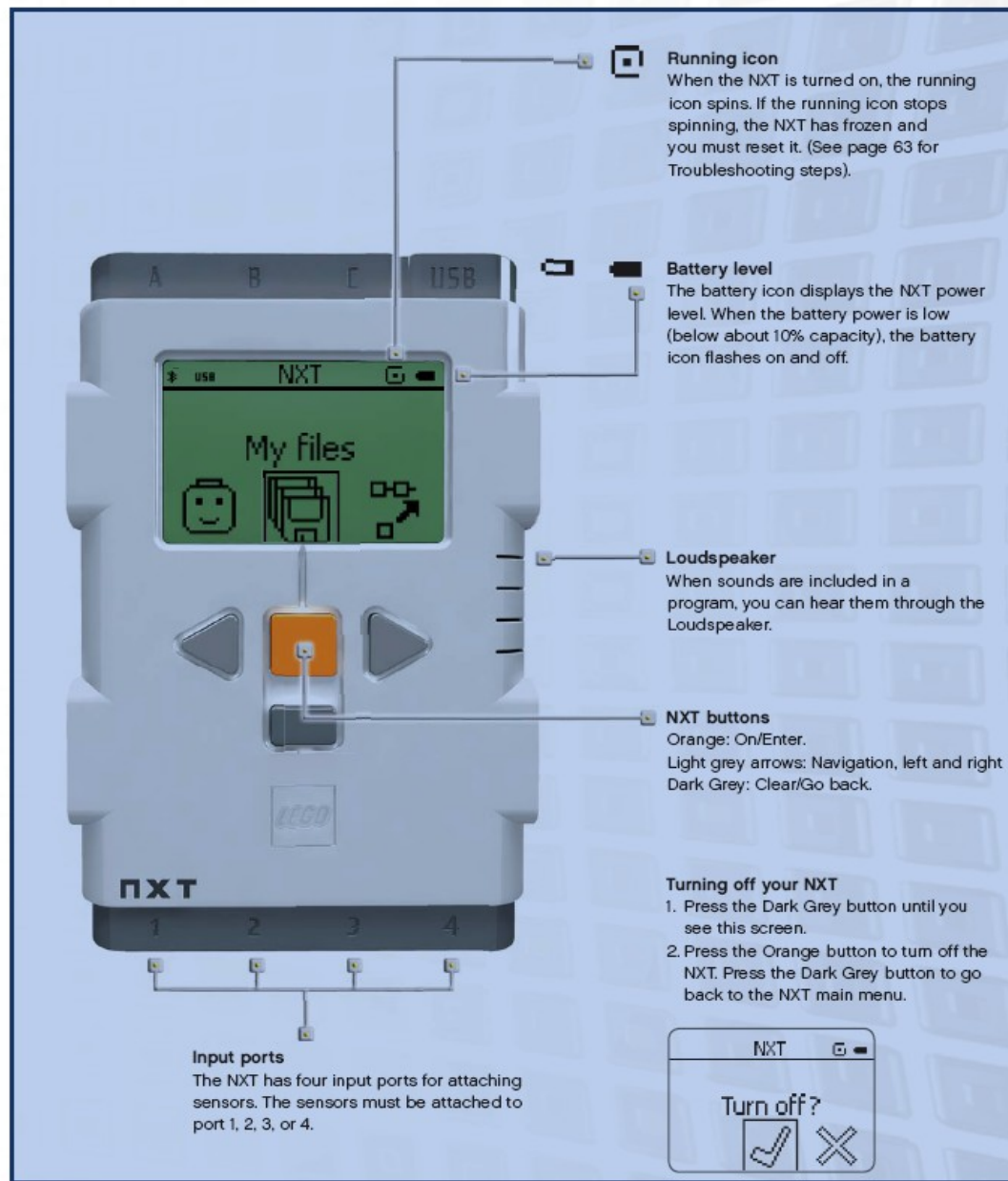
שימוש מעשי בחיישנים וניתוח מידע המופק מהם.

מהלך המעבדה

מעבדה זו מורכבת משני חלקים עיקריים: שליטה ברובוט ובמנועים, ושימוש בחיישנים.

מבנה בקר הרובולגו (brick)

ABOUT THE NXT BRICK



שימו לב שלבקר יש שתי קצוות.
בקצה אחד קיימות כניסות ממוספרות שאליהם יתחברו החיישנים.
ובקצה השני קיימות יציאות A,B,C שאליהם יתחברו המנועים.

חיבור פיסי של הרובולגו למחשב

בצע את החיבור בהתאם להנחיות [הקישור](#).

עקרונות עבודה עם מנועי הרובולגו

- הרובולגו מבצע את הפקודות בצורה אסינכרונית, לדוגמה אם הפקודה run נשלחת למנוע, המנוע יסתובב כל עוד לא ישלחו לו פקודה נוספת של brake.
- מנועים בסיבוב:
 - o הרצת מנוע יחיד, כשהשני כבוי.
 - o הרצת מנוע אחד קדימה, והשני אחורה.
- פקודות שימושיות לשימוש במנועים:
 - o Run(power)

המנוע יסתובב, ערך הפרמטר power יכול לנוע בטווח +100-, כאשר – המנוע יסתובב אחורה.
 - o ()Brake

המנוע יעצור.
 - o turn(power, dis4Turn)

המנוע יסתובב בכוח power, את מספר הסיבובים שמצוין ב-dis4Turn.
- עם run, עובדים בצמוד עם time.sleep, בעבודה עם turn אין לנו צורך בזה.

פקודות שימושיות לשימוש בחיישנים/אקטואטורים:

 - o <sensor_name>.get_sample()

קריאת ערך מחיישן, כאשר במקום <sensor_name>. יופיע שם המופע של החיישן.
 - o self.brick.play_tone_and_wait(262, 500)

self.brick.play_tone_and_wait(294, 500)

self.brick.play_tone_and_wait(330, 500)

self.brick.play_tone_and_wait(294, 500)

ביצוע "קטע מוסיקלי" בעזרת הרובולגו.

עבודה עם חיישנים:קישור לרובולגו ואיתחול החיישנים:

יש לשים לב לאיזה פורטים בפועל מחוברת החומרה.

קישור לרובולגו בשם שרה3

```
brick=nxt.locator.find_one_brick(name='Sara3')
```

הגדרת מנוע שמאלי וימני.

```
left=nxt.motor.Motor(brick,nxt.motor.PORT_A)
right=nxt.motor.Motor(brick,nxt.motor.PORT_B)
```

הגדרת חיישן מגע.

```
touch=nxt.sensor.Touch(brick,nxt.sensor.PORT_2)
```

הגדרת חיישן תאורה (מודד את מידת ההחזרה של האור).

```
light=nxt.sensor.Light(brick,nxt.sensor.PORT_3)
```

קביעת צורת הפעלת החיישן, ופורמט הנתונים המוחזרים

```
light.set_input_mode(nxt.sensor.Type.LIGHT_ACTIVE,
nxt.sensor.Mode.RAW)
```

משימות לביצוע:

עליך ליצור מחלקה שבעזרתה הרובולוגו:

1. ייסע ברבוע, כאשר כוח המנוע, זמן הנסיעה, ומרחק הסיבוב יקבל כפרמטרים.
 2. יבצע ספירלה ריבועית, כאשר הפרמטרים יהיו זהים לסעיף הקודם, בצרוף מספר צלעות הספירלה, ותוספת הזמן שיוצרת את הספירלה.
 3. יבצע מדידת כמות אור מוחזרת ממשטח שחור וממשטח לבן, וימצא את ערך הסף לכיוון החיפוש כך שיוכל לזהות מעל איזה פס הוא ממוקם.
 4. יספור את כמות הפסים השחורים תוך כדי נסיעה מעל "מעבר חצייה".
 5. ייסע קדימה ואחורה לסירוגין, תוך כדי השמעת מוסיקה.
- לפני ביצוע קידוד המחלקה אנא הקדישו זמן וחישובו על שיקולים תכנוניים כך שהקוד שלכם יהיה כמה שיותר יעיל ומדויק.
- שיקולים:

- אלו משתנים יופיעו כמשתני מחלקה, ואלו כמשתנים מקומיים.
- אלו פרמטרים תקבל כל מתודה.
- באלו מתודות של חבילת next-python כדאי לי להשתמש.
- אלו מתודות יופיעו במחלקה?

נסו לבצע משימות אלו לבדכם.

אם לא הצלחתם, תוכלו לקבל רמזים [מהסרטון הבא](#).

קובץ פייטון עם דוגמאות הקוד מהסרט נמצא [במודל](#).

הגשה:

יש להגיש ב-Verifier סרטונים של משימות 2,4, ומשימה 5.

בהצלחה!