

FLL Programming 101 NXT-G



Chapter #2
September 2007
Version 1.1b

Legal Stuff

© 2006 INSciTE in agreement with, and permission from FIRST and the LEGO Group. This document is developed by INSciTE and is not an official FLL document from FIRST and the LEGO Group. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by-nc-sa/2.0/>

or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

LEGO®, ROBO LAB, and MINDSTORMS™ are trademarks of the LEGO Group used here with special permission. FIRST™ LEGO® League is a trademark owned by FIRST (For Inspiration and Recognition of Science and Technology) and the LEGO Group used here with special permission. INSciTE™ is a trademark of Innovations in Science and Technology Education.

INSciTE

PO Box 41221

Plymouth, MN 55441

www.hightechkids.org

Creative Commons License

High Tech Kids is committed to making the best possible training material. Since HTK has such a dynamic and talented global community, the best training material and processes, will naturally come from a team effort.

Professionally, the open source software movement has shown that far flung software developers can cooperate to create robust and widely used software. The open source process is a model High Tech Kids wants to emulate for much of the material we develop. The open source software license is a key enabler in this process. That is why we have chosen to make this work available via a Creative Commons license. Your usage rights are summarized below, but please check the complete license at: <http://creativecommons.org/licenses/by-nc-sa/2.0/>.

Credits

This presentation was developed by Doug Frevert. It is based on the work of Fred Rose. The accompanying labs were originally done in RCX Code by Joel Stone and converted to ROBOLAB by Doug Frevert. A portion of the material is taken from *“Building LEGO Robots for FIRST LEGO League”* by Dean Hystad. Amy Harris defined the 10 programming steps. Eric Engstrom, Jen Reichow, and Ted Cochran reviewed ongoing drafts. Eric taught the first class and helped modify the content accordingly.

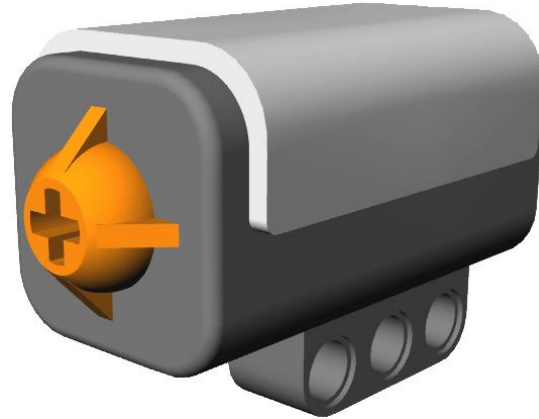
נתוני קלט

חיישנים

חיישנים

- אפשרו לרובוט שלכם לגלות את העולם האמיתי.
 - מגע
 - האם הרובוט שלכם יצר קשר עם משהו?
 - אור
 - האם פני השטח מוארים או חשוכים?
 - קול (מיקרופון)
 - אולטראסוניק (מרחק)
 - סיבוב
 - מובנה במנועים.
 - זמן
 - חיישן פנימי, עוקב אחר זמן.
 - מתח סוללה.

חיישן #1 - מגע



- כדי לזהות מגע או היתקלות במשהו.
- טוב לגילוי תנועות של זרועות הרובוט. החיישן פועל כאשר הזרוע זזה די רחוק על מנת לדחוף את חיישן המגע.

לחץ, משוחרר, נתקל במשהו

Port: 1 2 3 4

Action: Pressed Released Bumped

נתקל במשהו : נלחץ ושוחרר, בכל סדר שהוא.

בלוק ההמתנה עבור חיישן מגע

הרובוט מחכה עד שהחיישן מגיב.

לחץ, משוחרר, או נתקל במשהו.

Control: Sensor

Port: 1 2 3 4

Sensor: Touch Sensor

Action: Pressed Released Bumped

בלוק התנאי-חיישן מגע.

- המתנה עבור חיישן מגע יכולה להיות שימושית, אבל פעמים רבות אתה תרצה לעשות דברים שונים על בסיס הערך הנוכחי.

The image shows the Scratch 'Touch Sensor' block configuration window. The block is set to 'Sensor' control and 'Touch Sensor' sensor type. The 'Action' section has 'Pressed' selected. The 'Port' section shows ports 1, 2, 3, and 4. Annotations in Hebrew explain the port numbers:

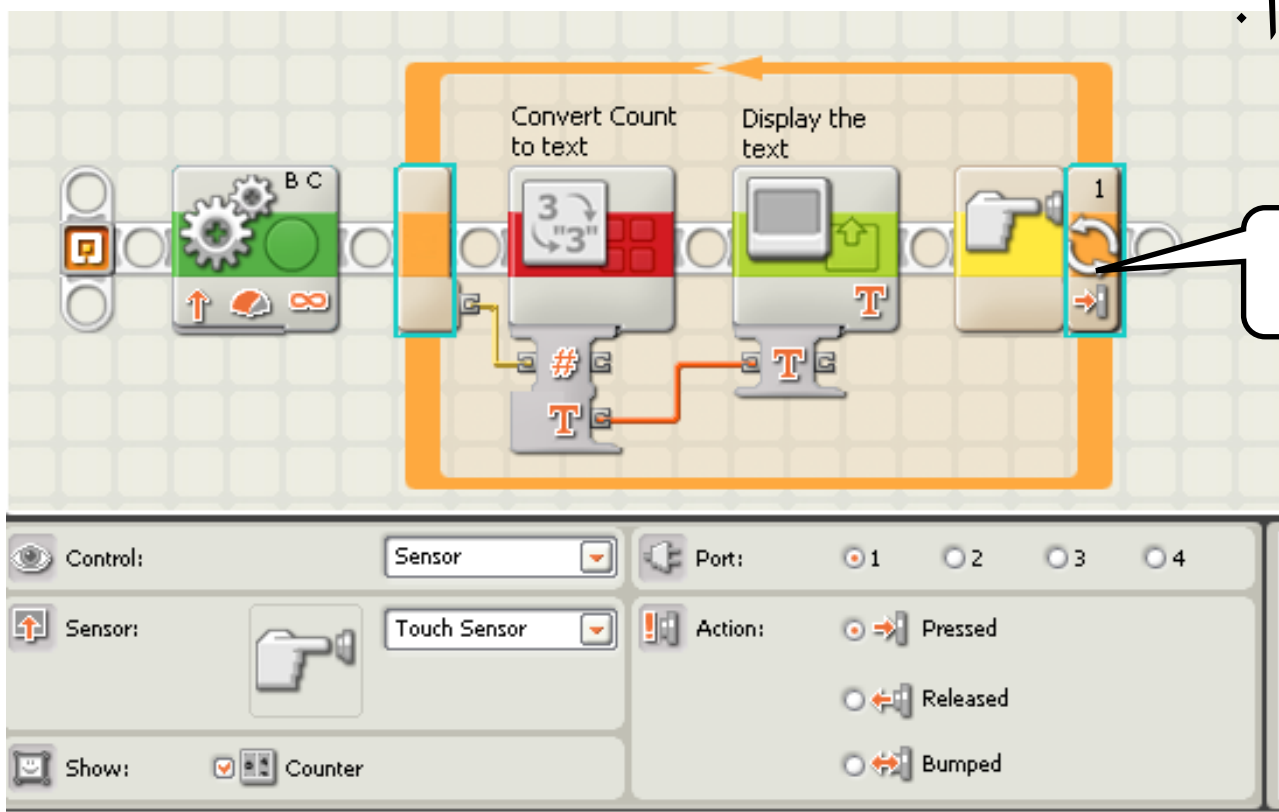
- אמת** (True): Points to the top half of the block, which is highlighted with a checkmark and the number '1'.
- שקר** (False): Points to the bottom half of the block, which is highlighted with an 'X'.
- יציאות חיישנים 1, 2, 3 או 4.** (Sensor outputs 1, 2, 3 or 4): Points to the 'Port' section, where port 4 is selected.

The background shows a Scratch stage with a 'Switch' block connected to a 'Touch Sensor' block. The 'Switch' block has two 'BC' (Binary Counter) blocks and one 'CB' (Counter) block. The 'Touch Sensor' block is connected to port 4 of the 'Switch' block.

פקודות של לולאת חיישן מגע

• לולאה עד שחיישן מגע לחוץ.

שימושי אם הלולאה מכילה פקודות שעשויות לחזור על עצמן.

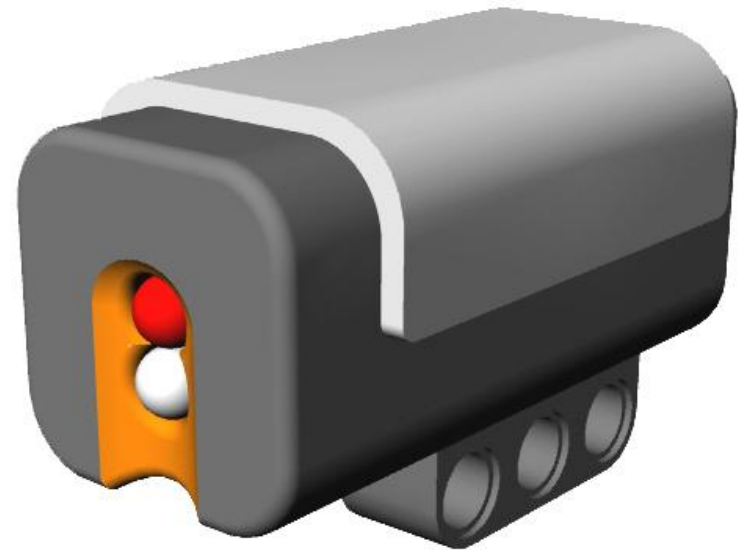


תוכן הלולאה תמיד רץ לפחות פעם אחת.

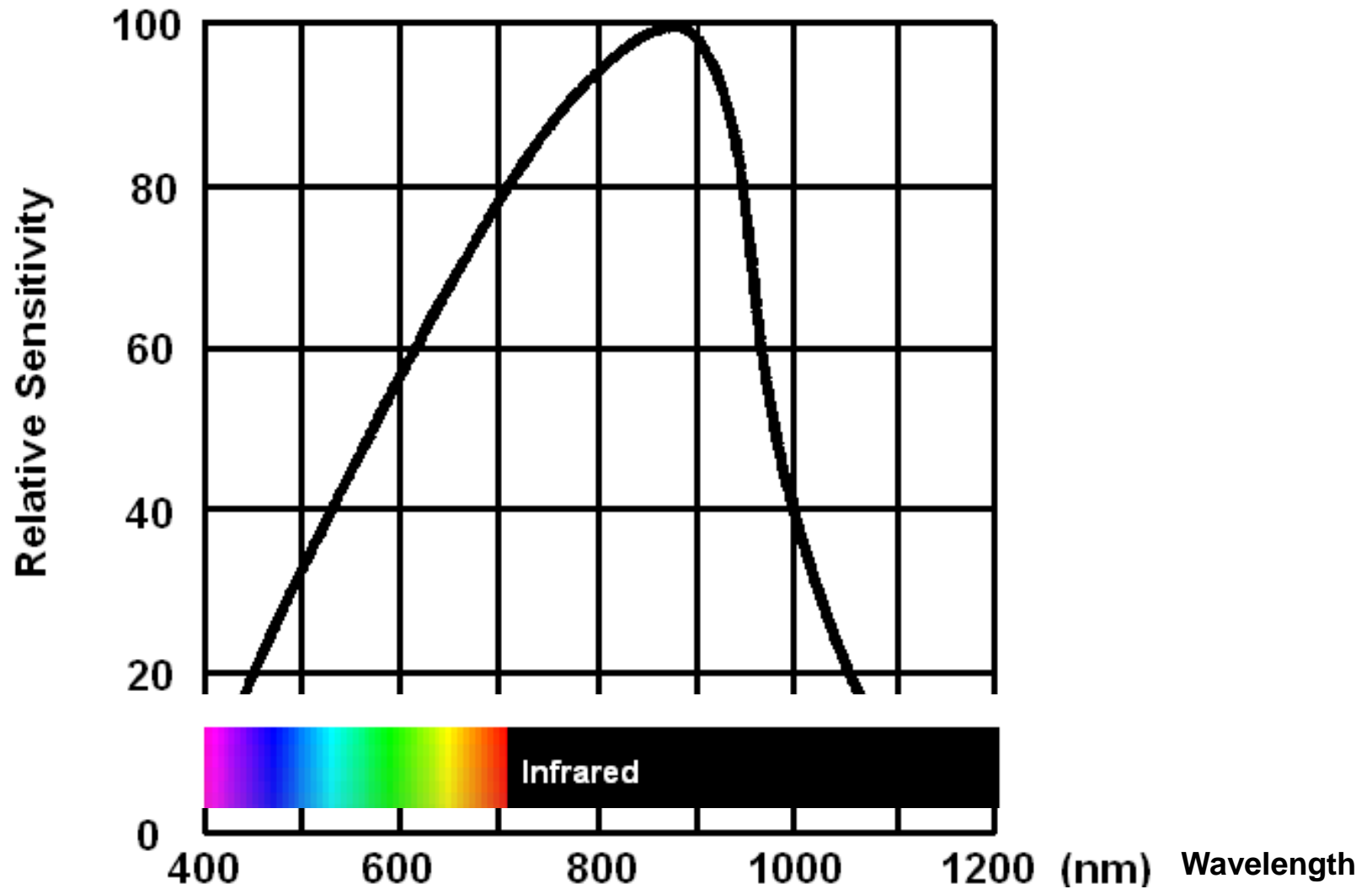
• למשל, רוטינה (שגרה) שמצפצפת עד שפגוש נתקל במשהו.

חיישן #2: אור

- פועל במצב של "אחוז"
- 0 עד 100
- מספר גבוה יותר = יותר אור .
- משטח מואר יותר מחזיר יותר אור.
- כיילו את החיישן.
- האור יכול להיות מכובה/כבוי.
- מקרין אור אדום.



ספקטרום חיישן האור

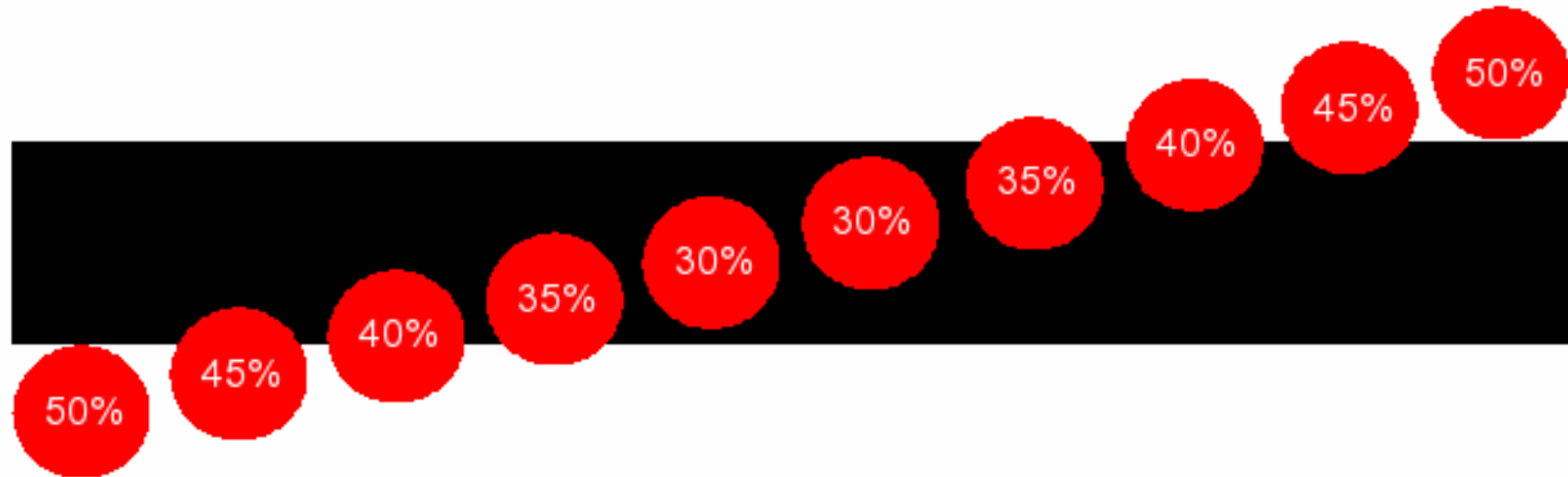


הכי רגיש לאור אדום/IR.

קריאות חיישן אור

- הקריאה הנמוכה ביותר – 5%.
- הקריאה הגבוהה ביותר – 100% (מצביע על אור).
- הקריאות תלויות גם בצבע פני השטח.
- ראה "בונים רובוטי-לגו עבור FLL, מאת דין היסטד.
- רגיש למרחק שבין החיישן ופני השטח מחזירי האור. וריאציות יכולות להפוך את הקריאות לבלתי-שמישות. שמרו על החיישן קרוב לפני השטח, אך לא קרוב מדי.
- מגנו את החיישן ממקורות אור אחרים.

קריאות חיישן אור



- חיישן האור ממצע את קריאותיו בערך מעל אזור מעגלי.
- אם תחצו קו מהר מדי, אתם עלולים לפספס אותו.
- בדקו וכיילו מחדש ביום התחרות.

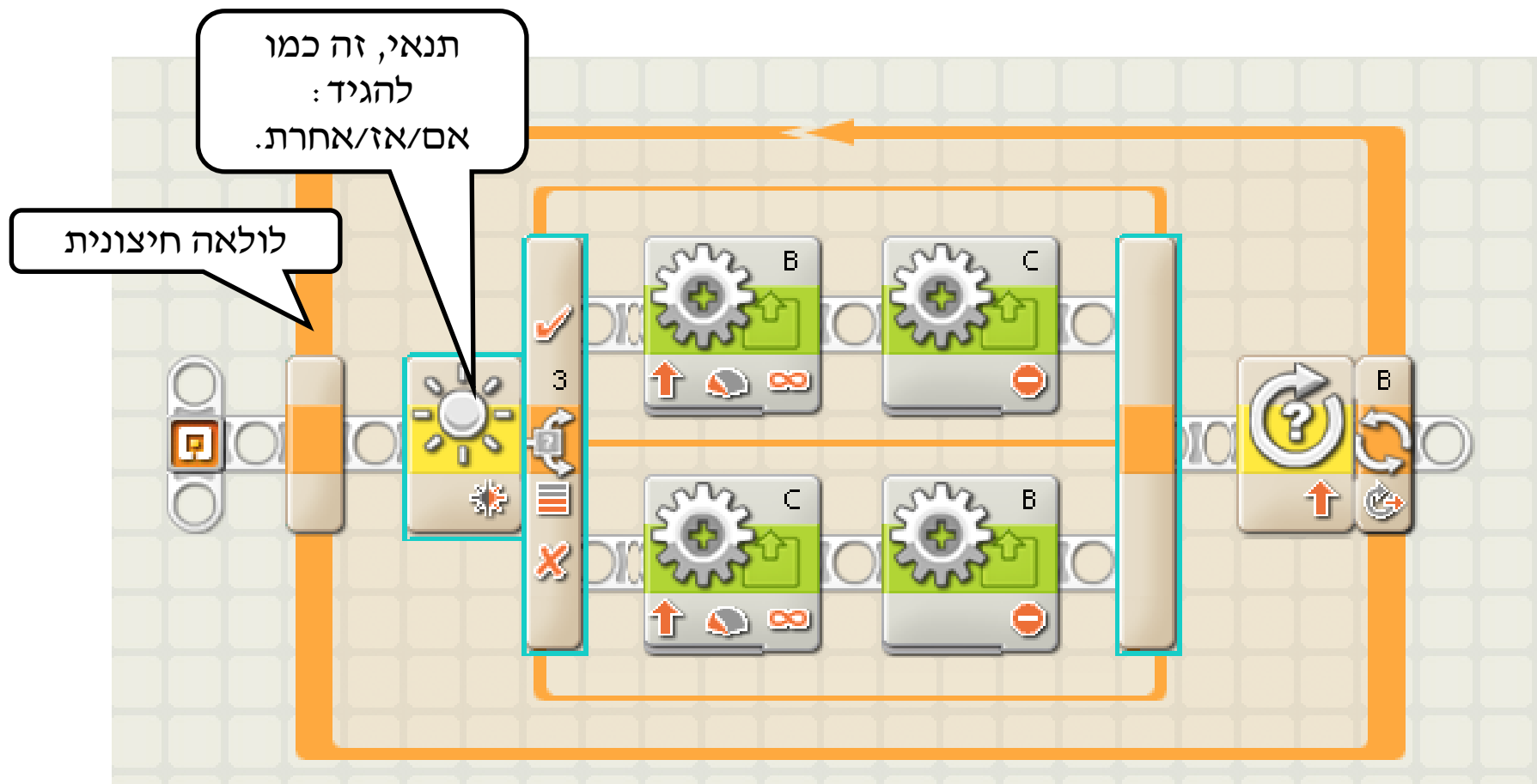
בלוק ההמתנה עבור חיישן אור

השתמש בבלוק "ההמתנה עבור.." כאשר צופים רק בחיישן אחד.

הדלק/כבה את האור

The image shows a programming interface with a sequence of three blocks: 'Forward', 'Right Turn', and another 'Forward'. The 'Right Turn' block is highlighted with a callout box. Below the blocks is a configuration panel for the selected block. The panel includes a 'Control' dropdown set to 'Sensor', a 'Port' dropdown set to '3', a 'Sensor' dropdown set to 'Light Sensor', an 'Until' slider, a 'Light' value of 40, and a 'Function' dropdown set to 'Generate light'.

בלוק התנאי-חיישן אור.

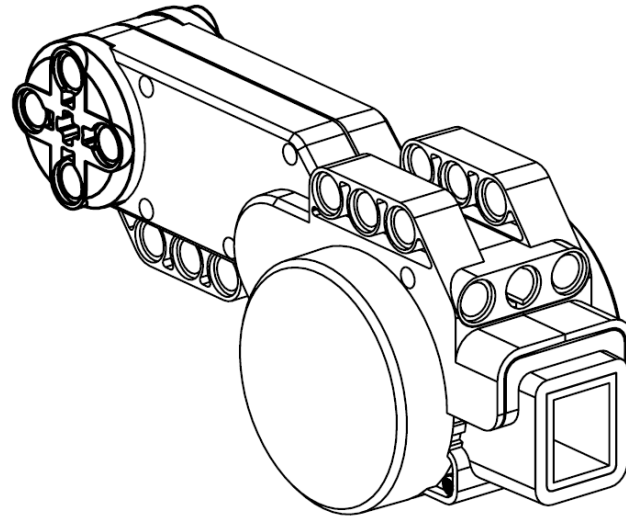


- אם בהיר יותר מ-50, הדלק את מנוע B. כבה את C.
- אחרת, הדלק את מנוע C. כבה את B. מה זה?

כיוול חיישן אור

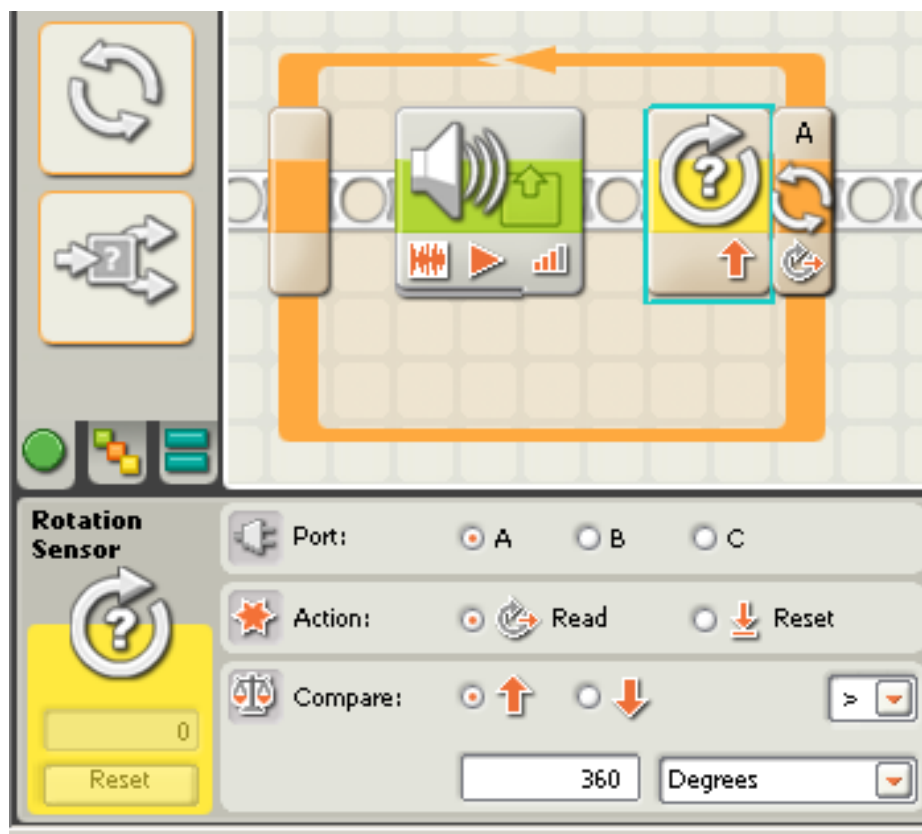
- הנע את הרובוט שלך על אזורים מוארים וחשוכים.
- מאפס אחוזי חיישן אור.
- יכול להיעשות גם ב-NXT.

חיישן #3: סיבוב



- מודד את כמות הסיבובים של ציר מסתובב. כאשר ציר מסתובב, ערכו של סופר הסיבובים ב-NXT גדל או קטן.
- 360 ספירות לסיבוב שלם.
- לכל מנוע יש חיישן סיבוב בתוכו.

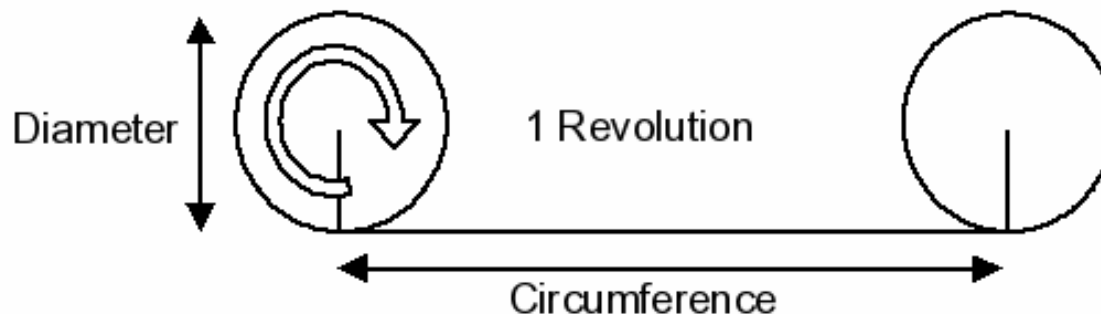
בלוק לולאת חיישן סיבוב



- ניתן לעשות לולאת תנאי סיבוב
- בחר כל לולאה. בחר "Sensor" ואז "Rotations".

חישוב מרחקים

- חיישן הסיבוב מאפשר גם לעשות חישובי מתימטיקה!

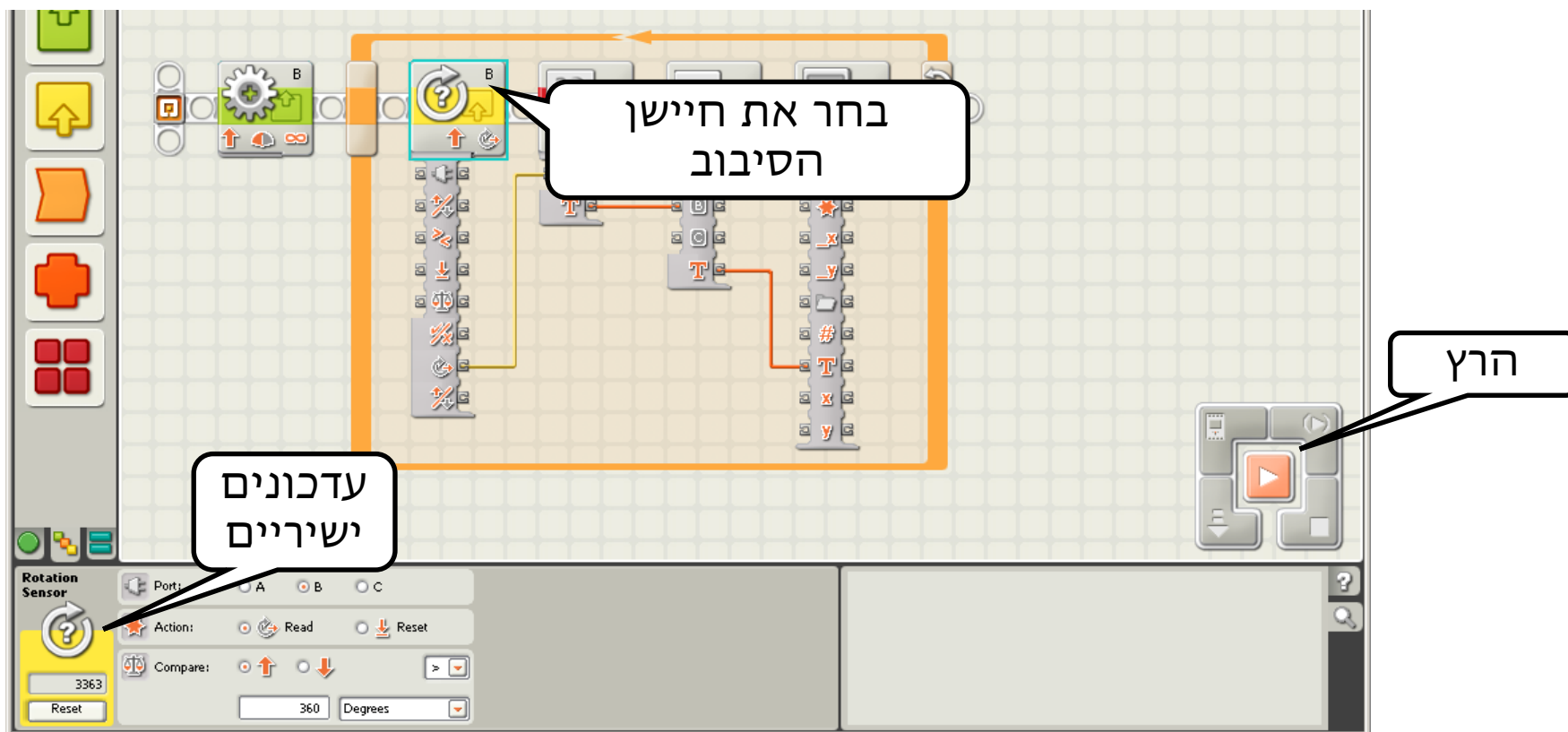


- כמובן שניתן גם למעול על פי שיטת גיסוי וטעיה.

- מקור הטעיות בחישובים – לכלוך על המשטח, שימוש במחלקים במקום גלגל, סידור לא טוב של ממסרים.

עוד על חיישני סיבוב

- חיישני סיבוב מודדים סיבובים קדימה וסיבובים אחורה
- עכדונים ישיריים מהרובוט למחשב (ע"י חיבור USB או Bluetooth).

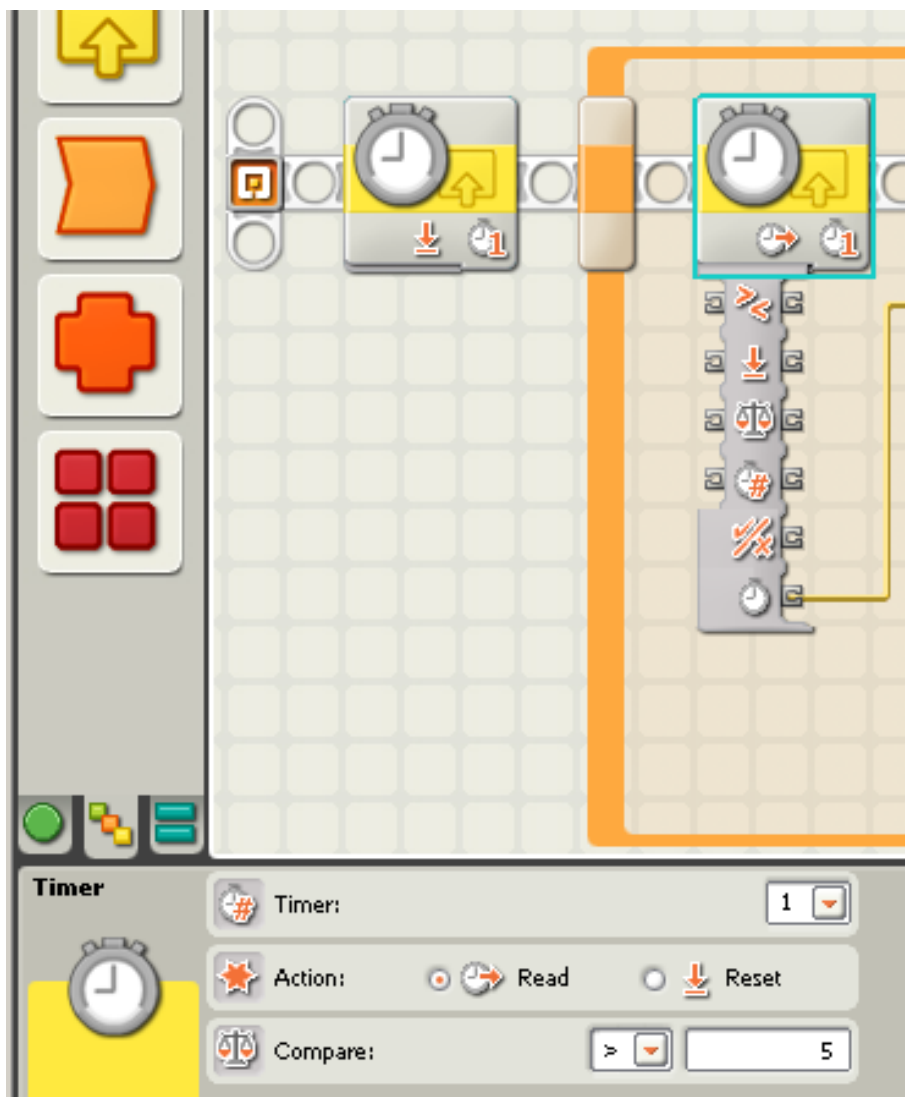


חקירה ותיקון שגיאות

• בעיות נפוצות

- **תכנות : אפסו את החיישנים לאפס לפני שמתחילים להשתמש בהם.**
- **חומרה : מנסים למדוד משהו במדוייק כאשר החיישן עצמו לו מדוייק.**
- **שליטה : התחלה, עצירה, סיבובים מהירים מדי.**
- **שינויים בתנאים מקדימים : שוכחים לסדר את הדברים כפי שהיו לפני שמריצים.**

חיישן #4: מודד-זמן



3 מודדי זמן

- זמן באלפיות השנייה
- בדיקות תנאי גדול מ- או קטן מ-
- אתחול
- נקודת הפעלה
- מספר מודד זמן
- כן לא
- זמן כמספר

חיישנים נוספים



- קול(מיקורופון)
- תחרות ה-FLL רועשת מדי?
- על-קולי(מרחק)
- התנגשות עם עוד NXT?
- כפתורי NXT
- לחיצה אחת להפעלת התוכנה הבאה
- הודעות קלט

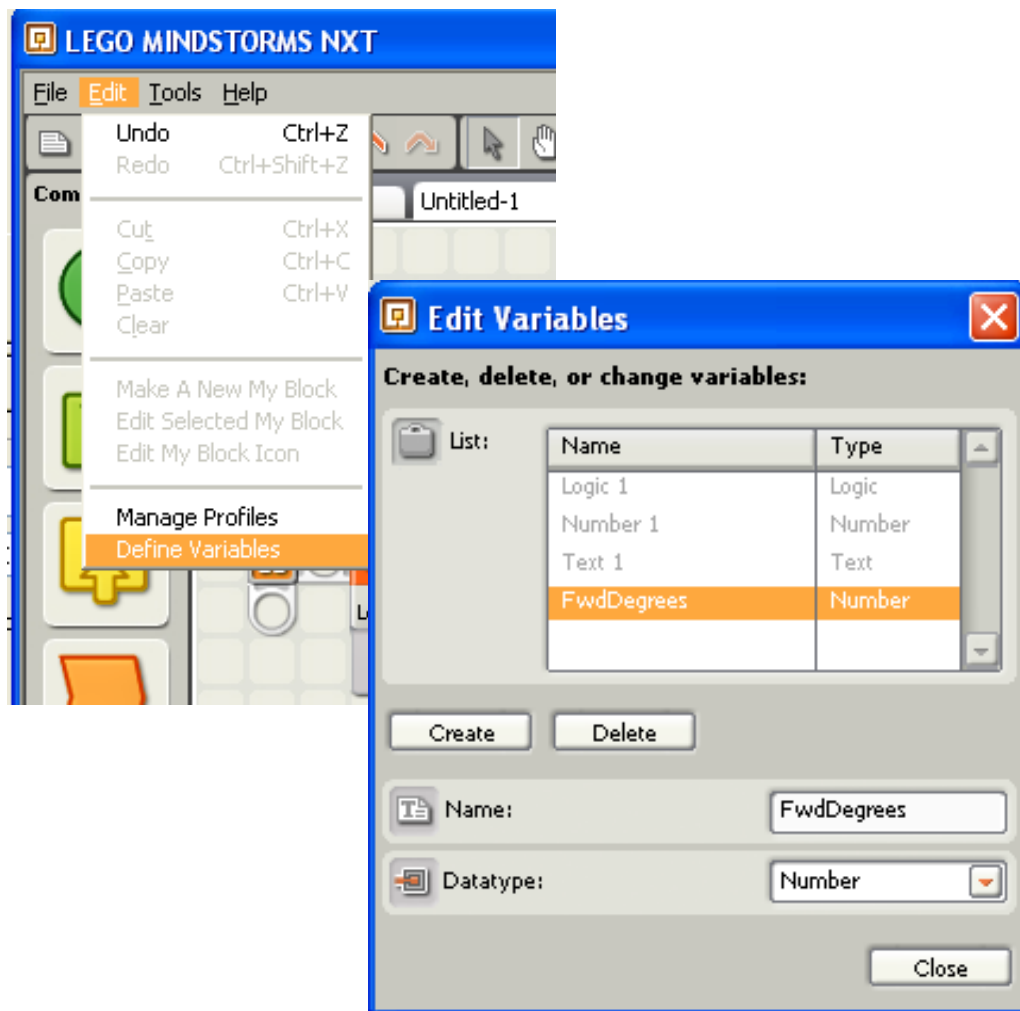
אסטרגיית KISS **Keep It Simple Strategies**

KISS #4: משתנים
#5: פעולות במקביל

מהו משתנה?

- ערך שניתן לשנות אותו במהלך התוכנית
 - הערך הזה "משתנה", על כן שמו נקרא כך.
- לדוגמה, התוכנית שלכם יכולה להכיל ערך של חיישן אור בזמן מסויים בתוך משתנה ששמו *LightBright*.
 - השתמשו במשתנה הזה מאוחר יותר.
- השתמשו בשם בעל משמעות מסויימת.
- שימושי להעברת ערכים לבלוק שלי.

KISS #4: משתנים



יצירת משתנים

<Edit><Define Variables>

Create

תנו שם למשתנה

בחרו סוג נתונים

מספר

טקסט

לוגי

שימוש במשתנה

השתמשו בפעולת הכתיבה

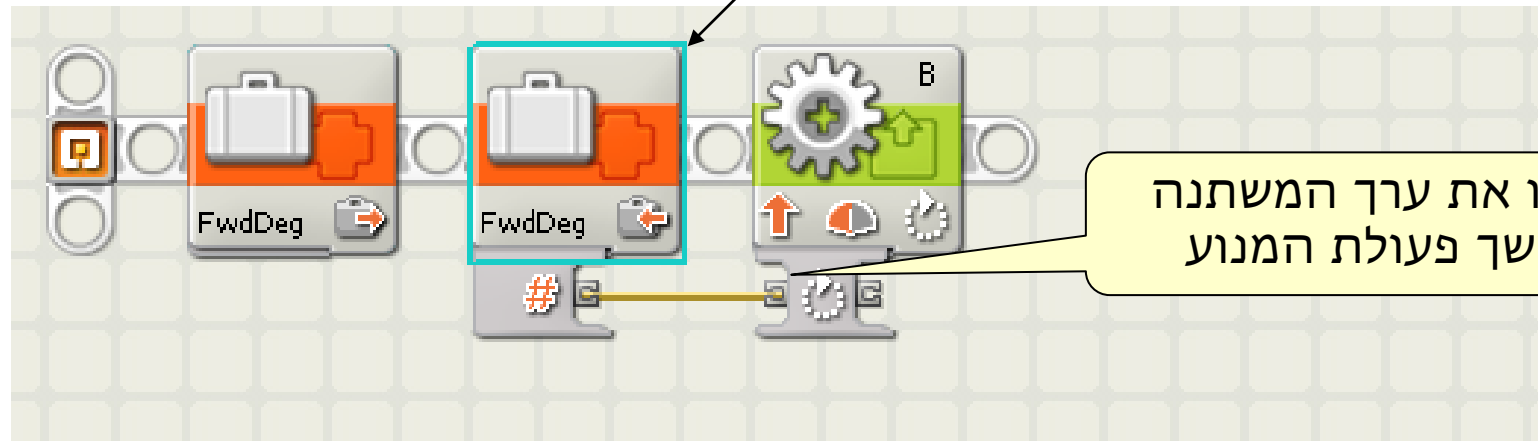
Name	Type
Logic 1	Logic
Number 1	Number
Text 1	Text
FwdDeg	Number

בחרו משתנה

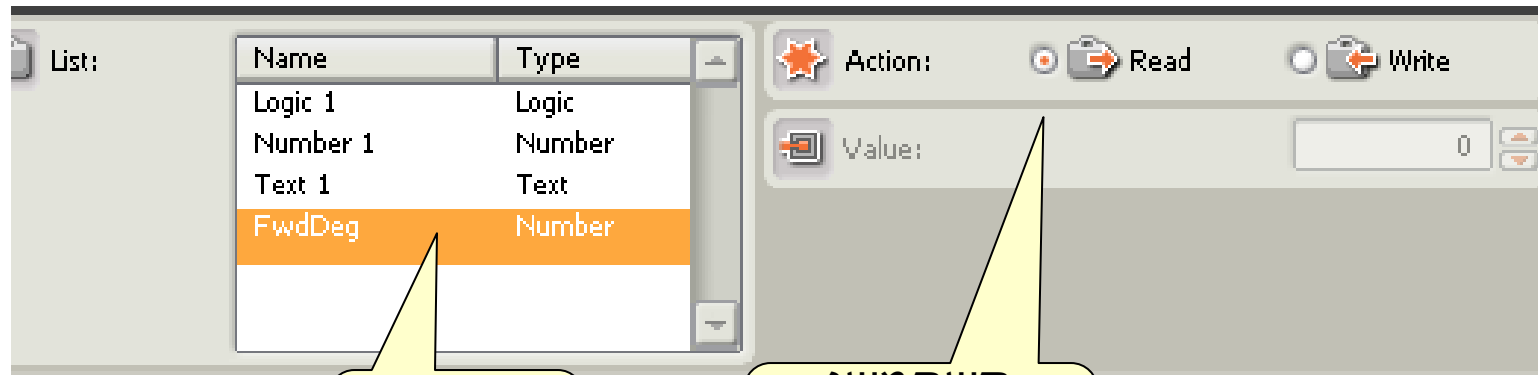
קבעו ערך

Value: 321

שימוש במשתנה



חברו את ערך המשתנה למשך פעולת המנוע



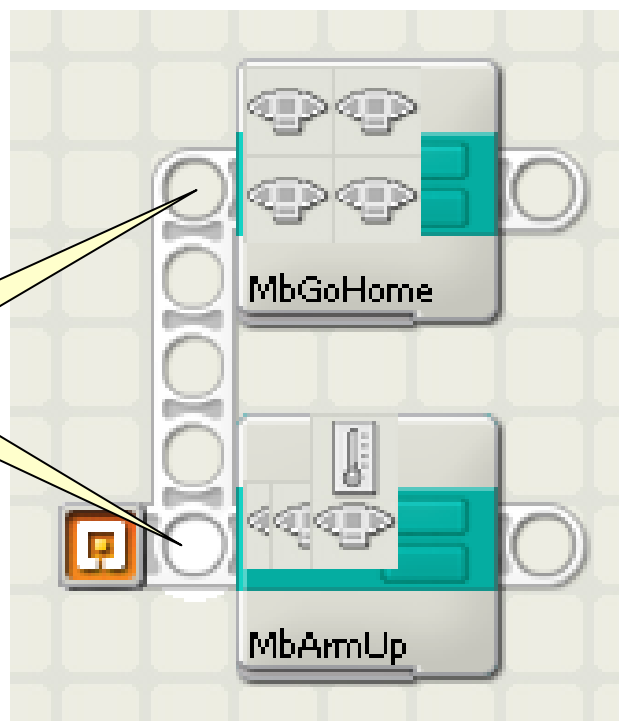
בחרו משנתה

השתמשו בפעולת הקריאה

KISS #5: פעולות במקביל

העבירו את הסמן שלכם לכאן. לחצו על כפתור ה-Shift והסמן יהפוך לסמן חיבור

גרו קו פעולה לכאן, ושחררו את העכבר.



- צרו שני פעולות הפעולות כל אחד בנפרד (אפשר ללכת ולשרוק בו זמנית?)
- פעולה אחת מעלה את הזרוע.
- פעולה שנייה חוזרת הביתה.



מעבדה שלוש

מטלה:

השתמשו בחיישן האור לגלות את הקווים הבאים.
 נועו מ-A ל-B ל-C ולבסוף ל-D.
 בונוס: השתמשו בחיישן המגע כדי לגעת בקירות
 ולעצור.



תשובה אפשרית למעבדה 3

The image displays a Scratch-style programming environment with a sequence of logic blocks. The top row consists of four logic blocks labeled 'A to B', 'Turn to C', 'B to C', and 'Turn to D'. Each of these logic blocks contains two sub-blocks: 'MbWait4Light...' and 'MbRightTurn'. An orange arrow above the top row indicates a loop from 'Turn to D' back to 'A to B'. Below the top row is a sequence of four blocks: 'Light_Pct', a timer block with '1', a speaker block, and a 'STOP' block. The bottom panel shows a variable list with 'Light_Pct' selected, and a control panel with 'Action' set to 'Write' and 'Value' set to 20.

Name	Type
Logic 1	Logic
Number 1	Number
Text 1	Text
Light_Pct	Number

Action: Read Write

Value:

נושאים מתקדמים

אמצעי תיקון שגיאות

החלפה בין תנאים

פעולות עם קבצים

התקנת תקשורת Bluetooth

מקורות נוספים

אמצעי תיקון שגיאות

• מוסיקה



- השתמשו במוזיקה כדי לדעת באיזה קטע קוד התוכנית נמצאת כעת.

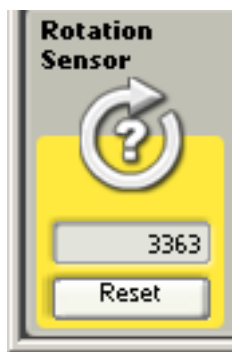
• LCD תצוגת



- כתבו טקסט, אפילו גרפיקה, למסך ה-LCD של ה-NXT.

• עדכונים ישיריים מה-NXT

- השתמשו ב-NXT-G כדי לקבל ערכים של משתנים וחיישנים על המחשב.



החלפה בין תנאים

בלוק סוויץ' עם 3 לשוניות, אחד עבור כל תנאי.

לחצו כאן להוספת תנאים.

מספר או טקסט.

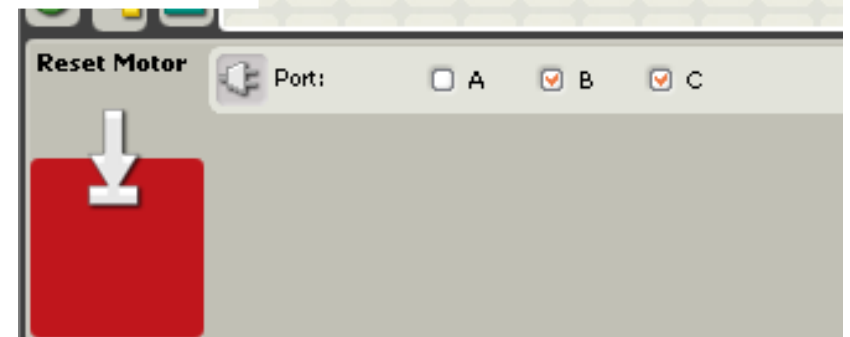
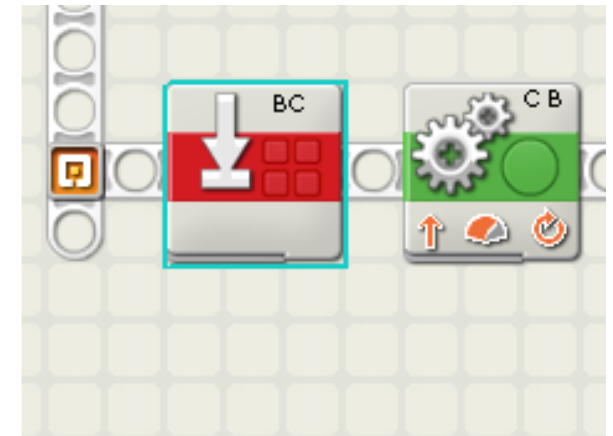
לא להשתמש ב-Flat view.

Number	Condition
1.	dark
2.	grey
✓ 3.	light

light

בלוק אתחול מנוע

- בלוק תנועה של מנוע הוא פקודה להזזת מנוע סרבו
- מתקן מומנט אחרי מספר רצות .

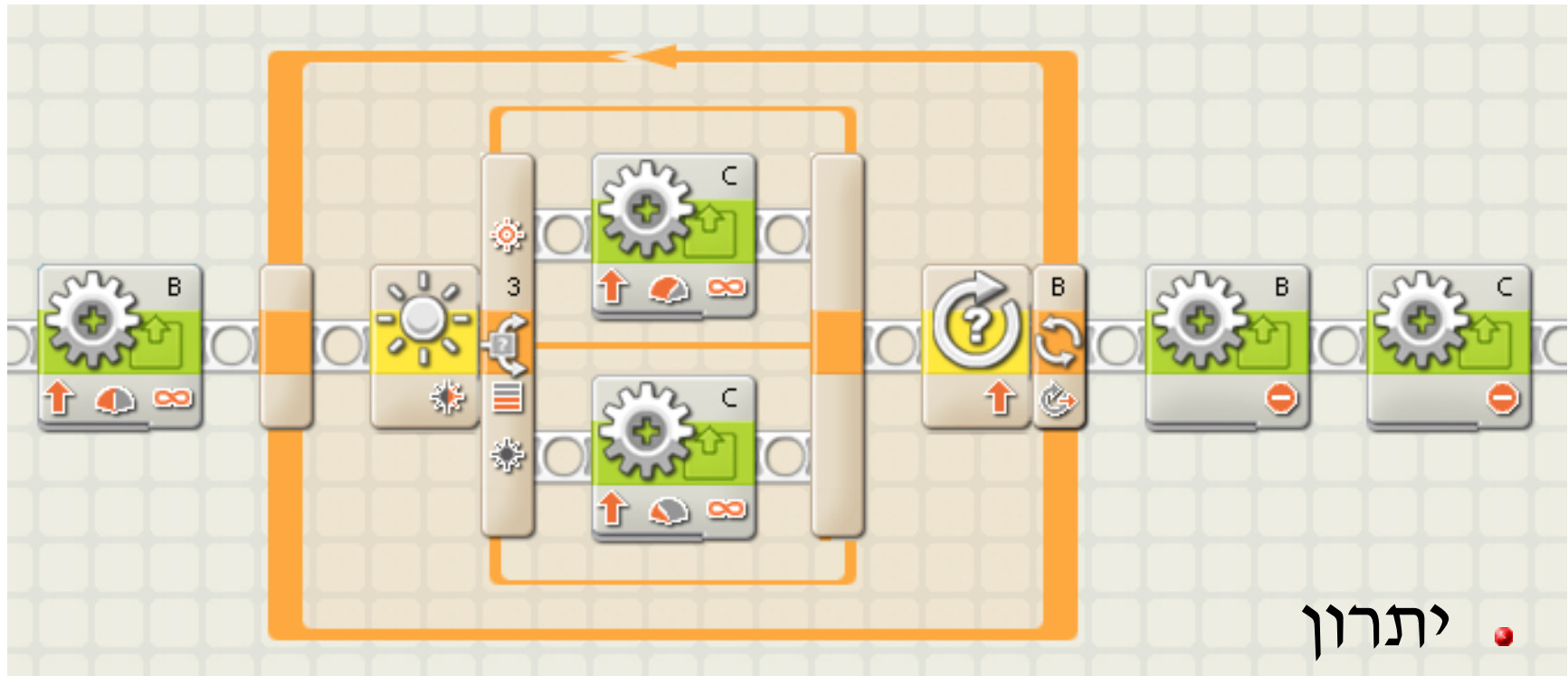


חשבון

The image shows a Scratch calculator script with four script blocks connected in sequence. The first block is 'FwdDeg' (Forward Degrees). The second block is 'Addition' (plus sign), which is highlighted with a yellow callout box labeled 'פלט' (Output). The third block is 'Repeat' (circular arrow with '3'), and the fourth block is 'Say' (computer monitor with speech bubble). A yellow callout box labeled 'קלט מחובר' (Input connected) points to the 'A' input field of the second block. A yellow callout box labeled 'פעולה' (Operation) points to the 'Addition' dropdown menu in the calculator interface below the script. The calculator interface shows 'Operation: Addition', 'A: 0', and 'B: 30'.

השוואת אלגוריתמים

השוו את קוד מעקב אחרי קו לעומת קוד זה:



פשוט. מהיר. רק מנוע אחד זז בכל פעם.

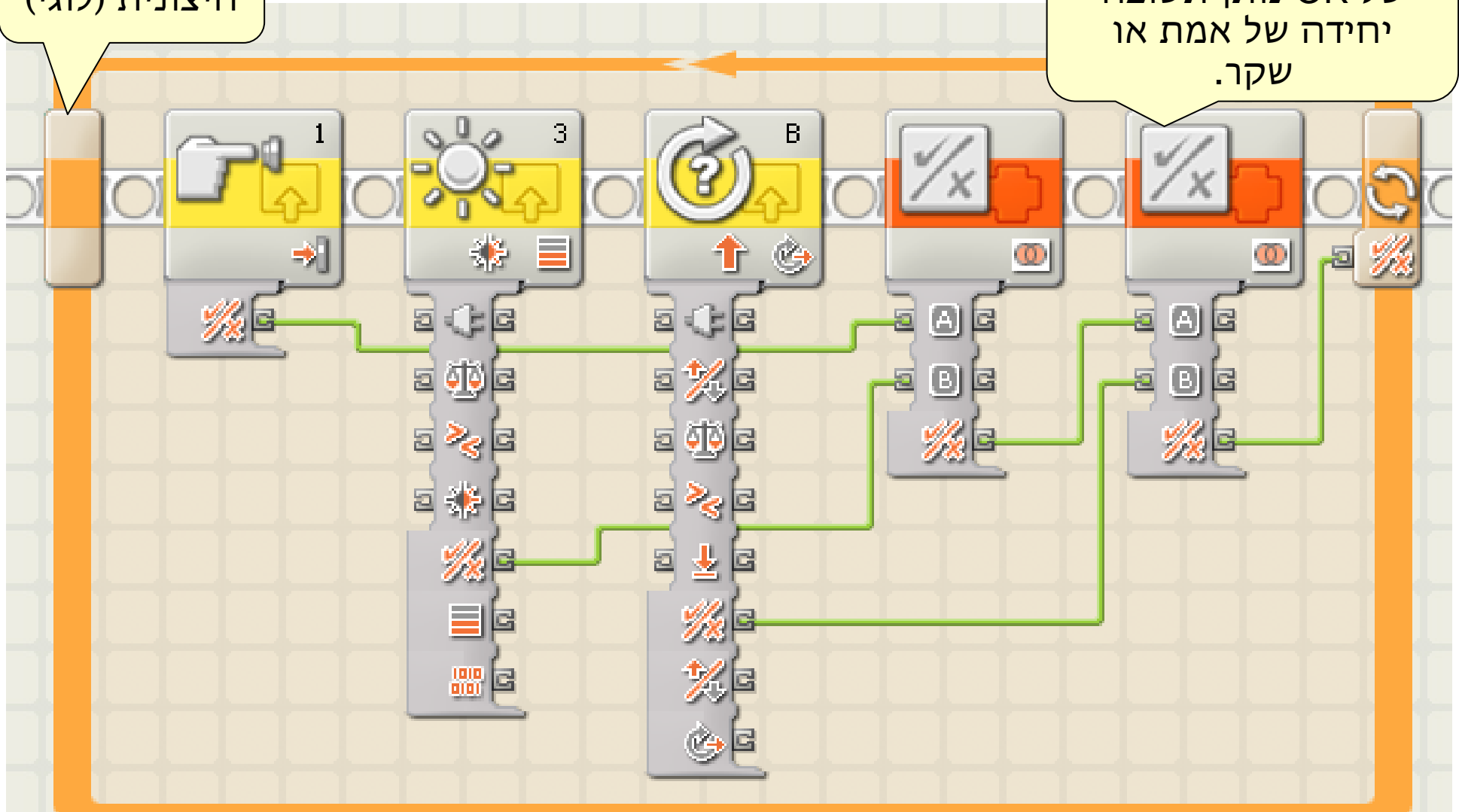
חסרון

פניות חדות?

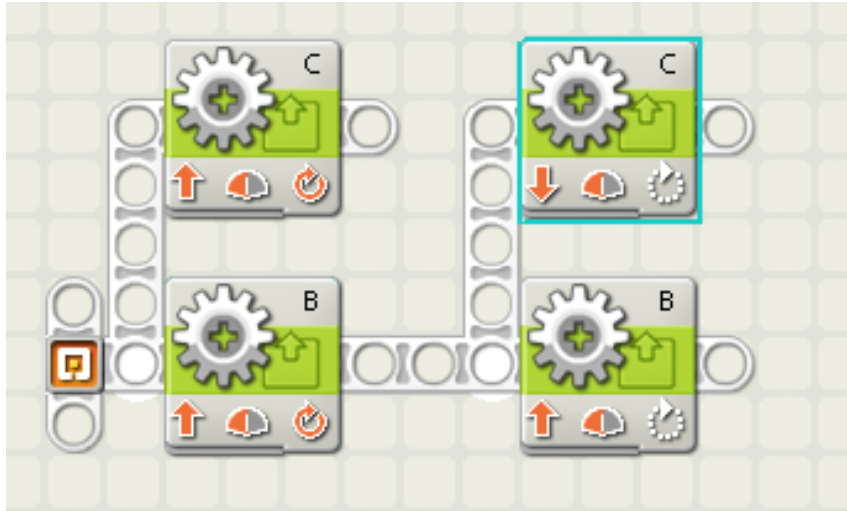
מעקב אחר 3 משתנים

לולאה
חיצונית (לוגי)

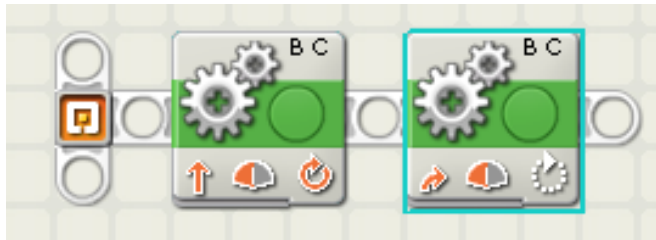
שילוב מספר של לוגים
של OR נותן תשובה
יחידה של אמת או
שקר.



יעול קוד

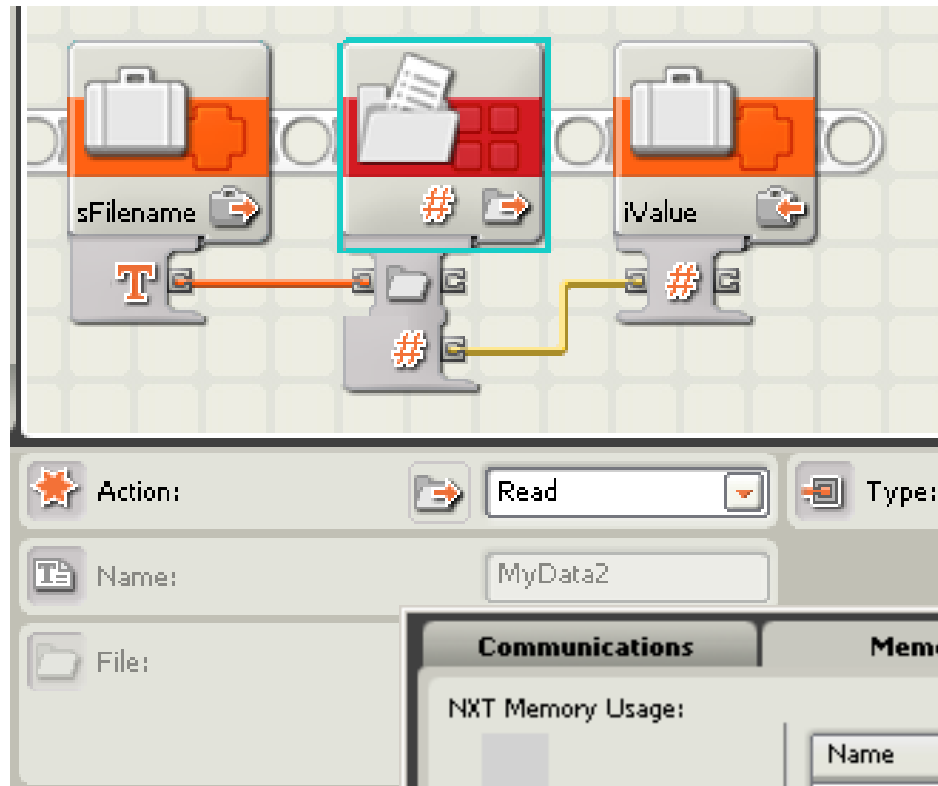


- איזה יותר מהיר?
יותר אמין?



- השתמשו בקוד שהכי היגיוני לכם, המתכנתים.

פעולות עם קבצים



- קריאה
- כתיבה
- סגירה
- מחיקה

Communications

NXT Memory Usage:

Delete All

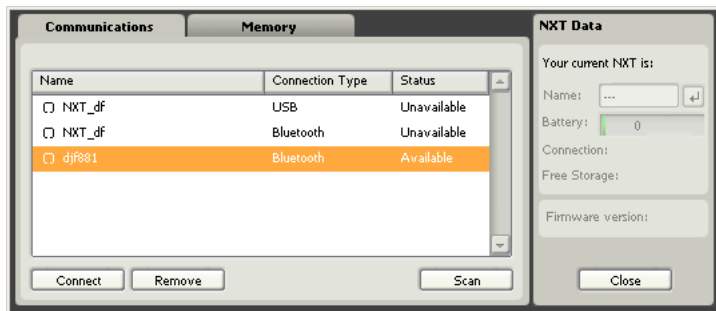
Memory

Show System Files

Name	Size
MyData3.txt	0.2 KB
MyData2.txt	0.2 KB
MyData.txt	4.0 KB

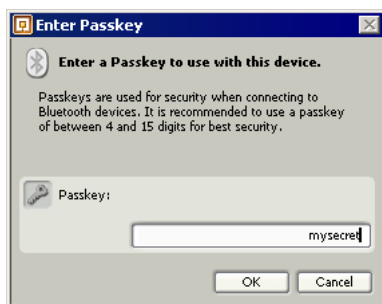
Upload Download Delete

התקנת תקשורת Bluetooth



- על ה-NXT - בחרו :

Bluetooth -> On/off -> ON
 Bluetooth -> Visibility -> Visible



- ב-NXT-G בחרו ב-NXT ולחצו על "Connect"

- הקלידו סיסמה, כגון "סיסמהשלי" ולחצו

על OK.

- על ה-NXT יפיוע מסך. עליכם להקליד את אותה סיסמה מהר (יש לכם 30 שניות).

- אחרי שה-NXT-G מראה "Connected",
 Bluetooth -> Visibility -> Invisible

מעבדה ארבע

מטלה:

נועו סיבוב אחד שלם מסביב למסלול מעגלי.
(קו שחור של 2 ס"מ על דף לבן)

פתרון אפשרי למעבדה 4

