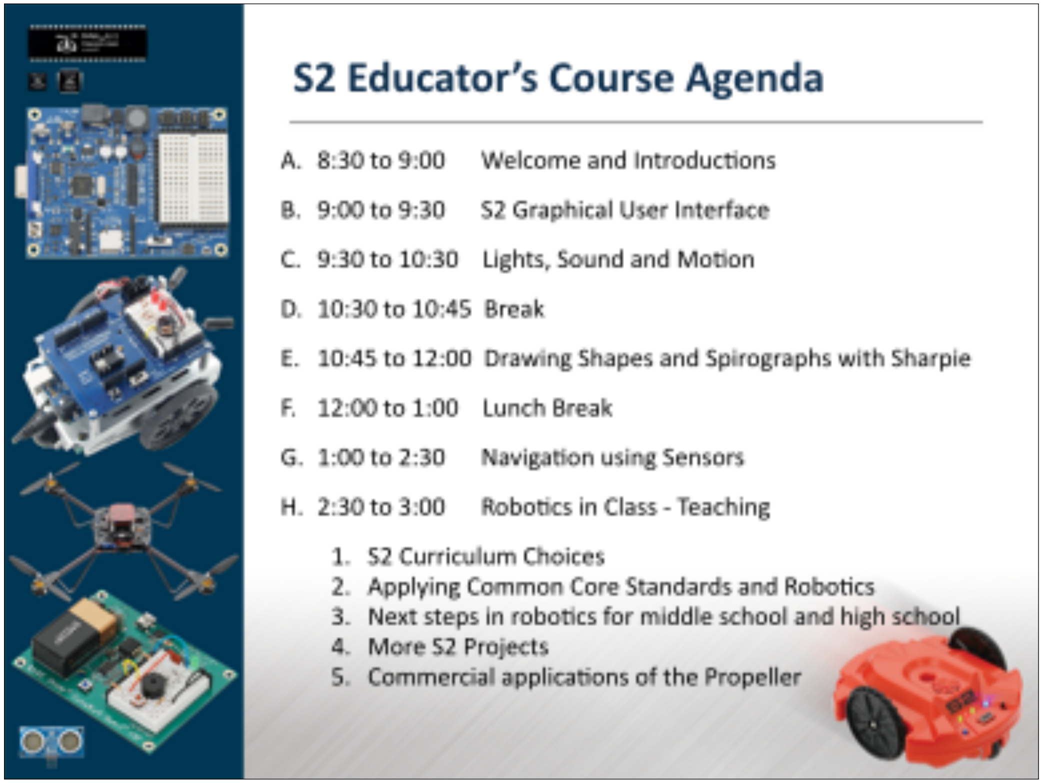
S2 Robot Educator's Course

Ken Gracey, CEO
Parallax Inc.

kgracey@parallax.com
Direct (916) 625-3010



Introductions and software installations.
Robotics is the ideal combination of audio, visual and tactile learning.
Careers in engineering are influenced by what students see when they are young.
When interest and careers combine positive results occur!
Loading the programs and S2 GUI is the first order of operation today.

Thoughts to keep in mind!

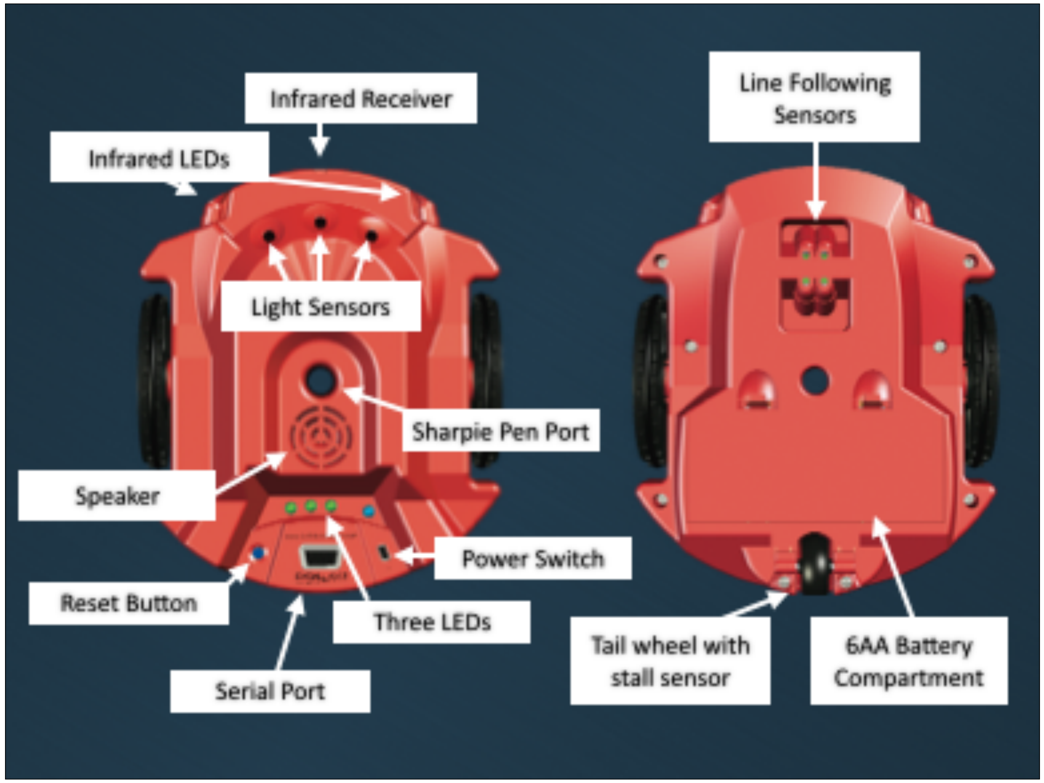
Sharing with students is often more important than teaching them.

The most successful careers happen when personal and professional interests collide!

We are using graphical programming, the first step towards writing code in BASIC, C, C++. The concepts used will make it easier to envision variables, subroutines, and program organization in written code.

This class is fun professional development. Let's enjoy each other's company, share our experiences, and learn together.

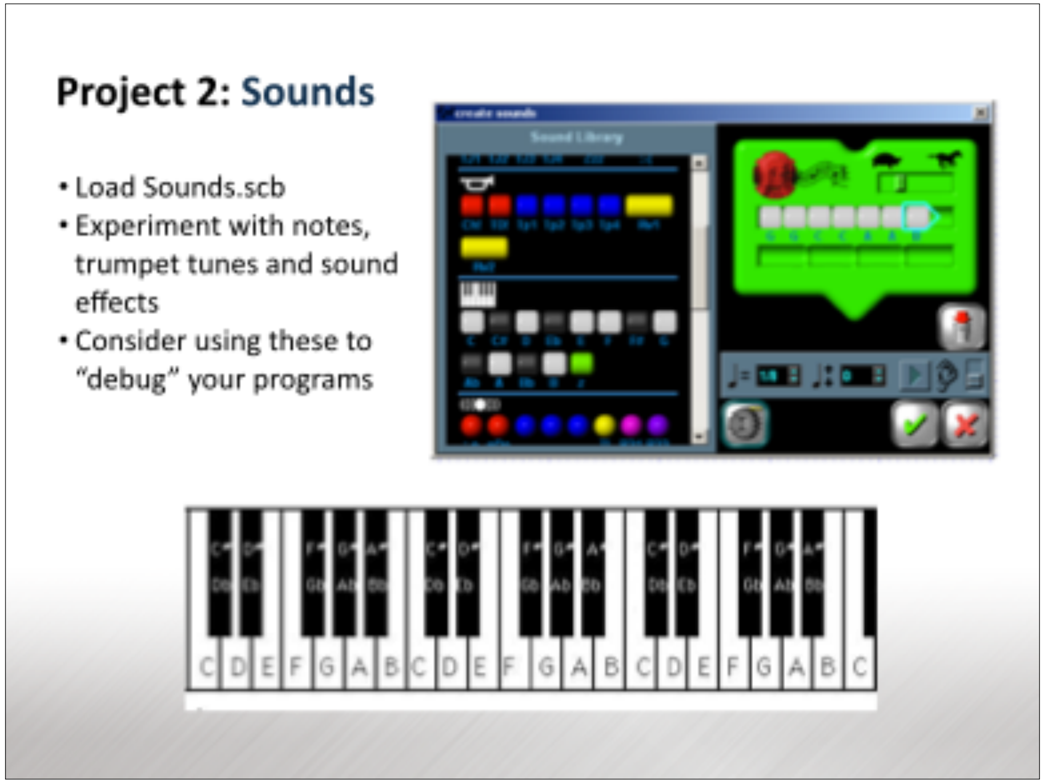
Some students who do poorly in other subjects are truly inspired by robotics.



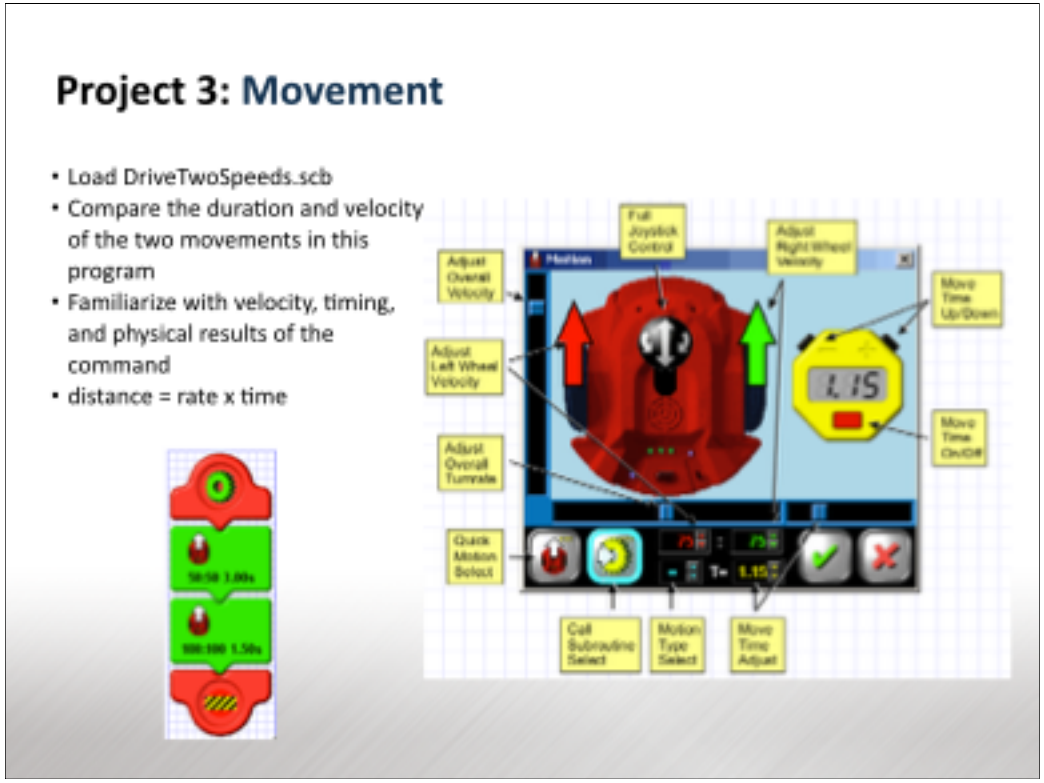
Many open hacks are available on-line: pin lifters, rubber band guns, etc.

This program is equivalent of a “Hello World” download to a microcontroller.
How long does it take for one loop to execute?
What percentage of the total time are the two outer LEDs illuminated?
See also the Neil Rosenberg “Robots for Beginners” Project 1: Light those LEDs.
CCSS.MATH.CONTENT.5.NBT.A.1 Recognize that in a multi-digit number, a digit in one place represents 10 times as much as it represents in the place to its right and 1/10 of what it represents in the place to its left.

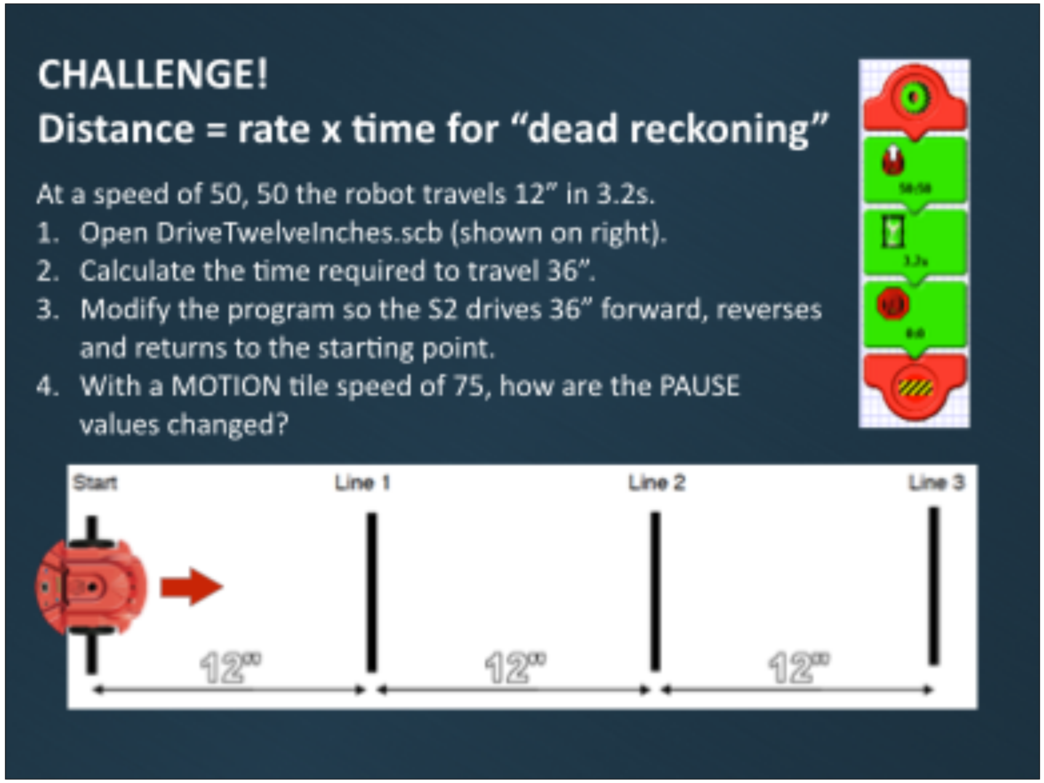
LEDsCylonChaser.scb provides an example to solving this challenge.



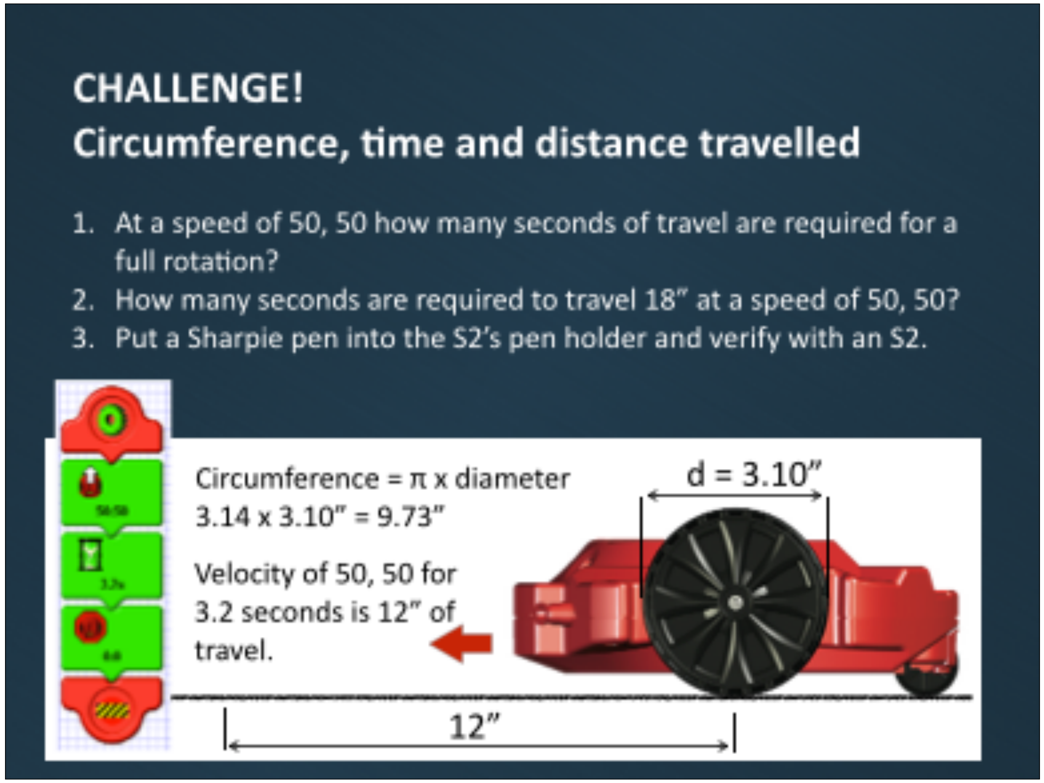
Debugging programs is a methodical way of determining what your program is doing.
Since robots of this type roll away and are not connect to your PC, you rely on visual and audio feedback.
See also the Neil Rosenberg "Robots for Beginners" Project 2: Sound OFF (and ON)
CCSS.ELA-LITERACY.SL.6.5 Include multimedia components (e.g., graphics, images, music, sound) and visual displays in presentations to clarify information.



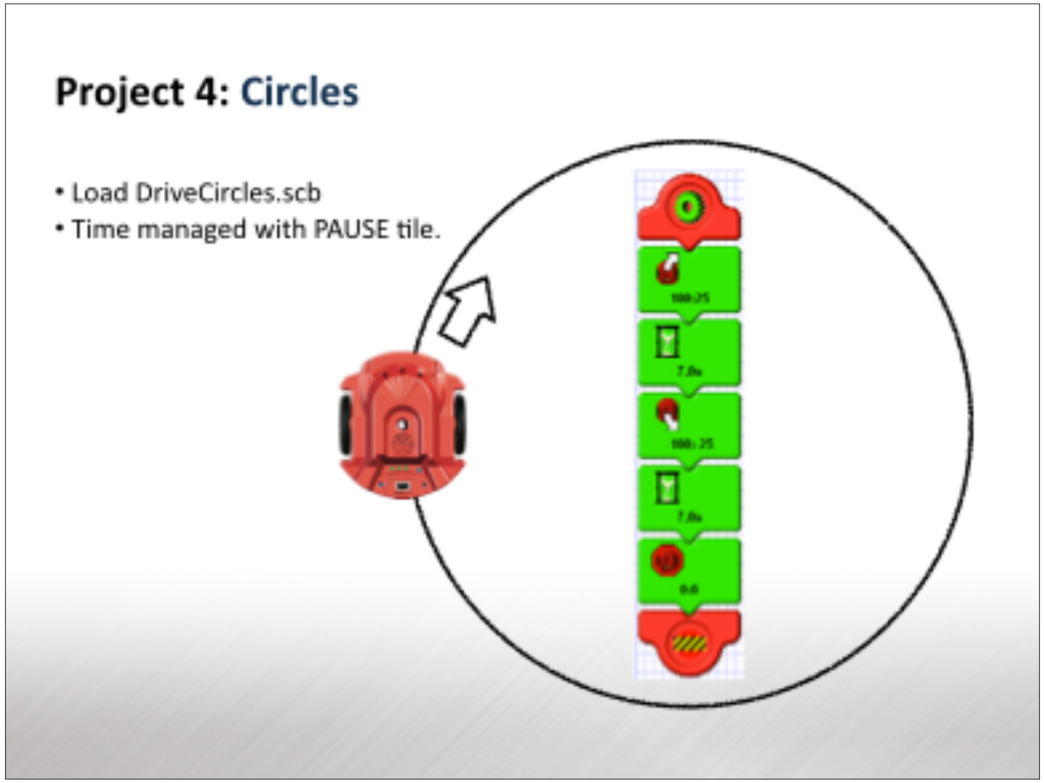
Discuss ways to increase time: loops, providing 0 as a movement time
Program the S2 to create circles, squares and shapes
Create a simple maze pattern; ask students to navigate it without sensors.
This is an example of open-loop robotic programming.
See also the Neil Rosenberg “Robots for Beginners” Project 3: Straight on till Dawn.
CCSS.MATH.CONTENT.8.EE.B.5 Graph proportional relationships, interpreting the unit rate as the slope of the graph. Compare two different proportional relationships represented in different ways. For example, compare a distance-time graph to a distance-time equation to determine which of two moving objects has greater speed.
CCSS.MATH.CONTENT.4.MD.C.6 Measure angles in whole-number degrees using a protractor. Sketch angles of specified measure.



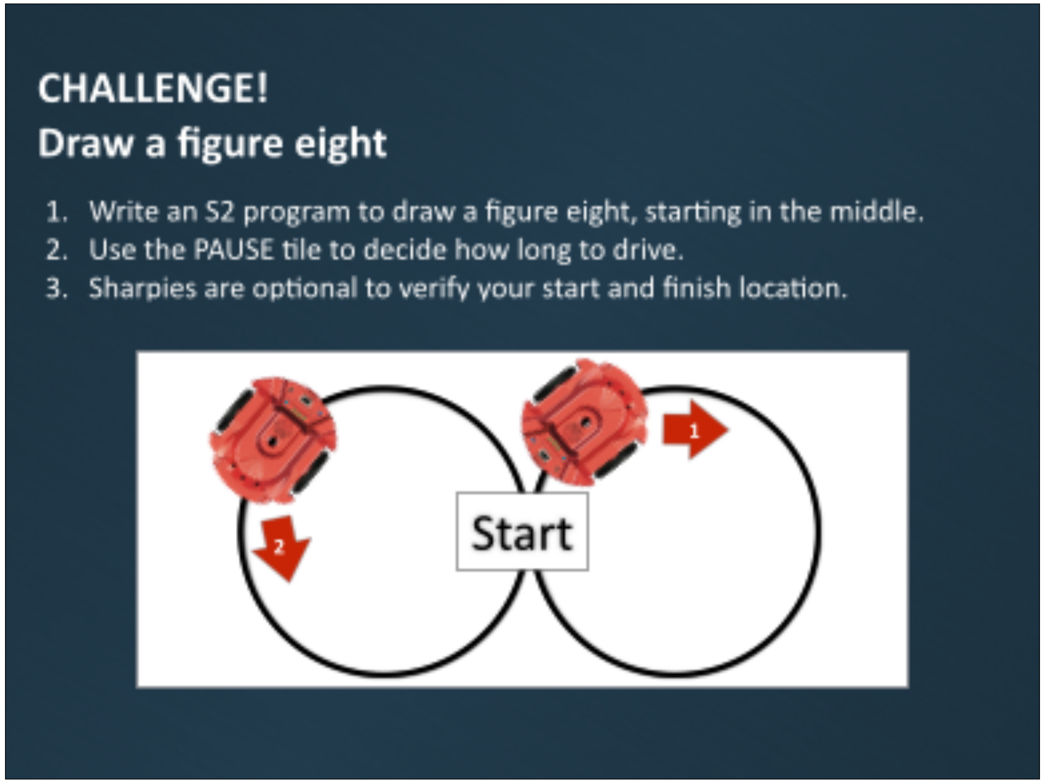
This program introduces the PAUSE tile and uses zero as a timing constant in the MOTION tile.
Many variations of this specific "distance = rate x time" variations are possible.
Given a maze, open-loop (no sensors) feedback, students can calculate the program faster than they can develop it from iteration.
See also "Applying Common Core Standards in Grades 4th-10th Using LEGO Robotics" http://nia.ecsu.edu/reuomps2012/teams/math/report%20draft17.pdf and "Using Robotics to Promote Learning in Early Grades" http://www.asee.org%2Fpublic%2Fconferences%2F8%2Fpapers%2F5480%2Fdownload&ei=MLY-VJ78EcWZ8gHvxIGoDg&usg=AFQjCNGcHZJ_tCoB1-utqnCH1F2876wajg&sig2=-tm1L7i3c6bZCIXsHuRfHg&bvm=bv.77412846,d.b2U
CCSS.MATH.CONTENT.8.EE.B.5 Graph proportional relationships, interpreting the unit rate as the slope of the graph. Compare two different proportional relationships represented in different ways. For example, compare a distance-time graph to a distance-time equation to determine which of two moving objects has greater speed.



Divide the circumference into 12” to obtain seconds, or 9.73” / 12” = 0.81. Multiply by 3.20 seconds = 0.81 x 3.20s = 2.60s.
CCSS.MATH.CONTENT.7.G.B.4 Know the formulas for the area and circumference of a circle and use them to solve problems; give an informal derivation of the relationship between the circumference and area of a circle.



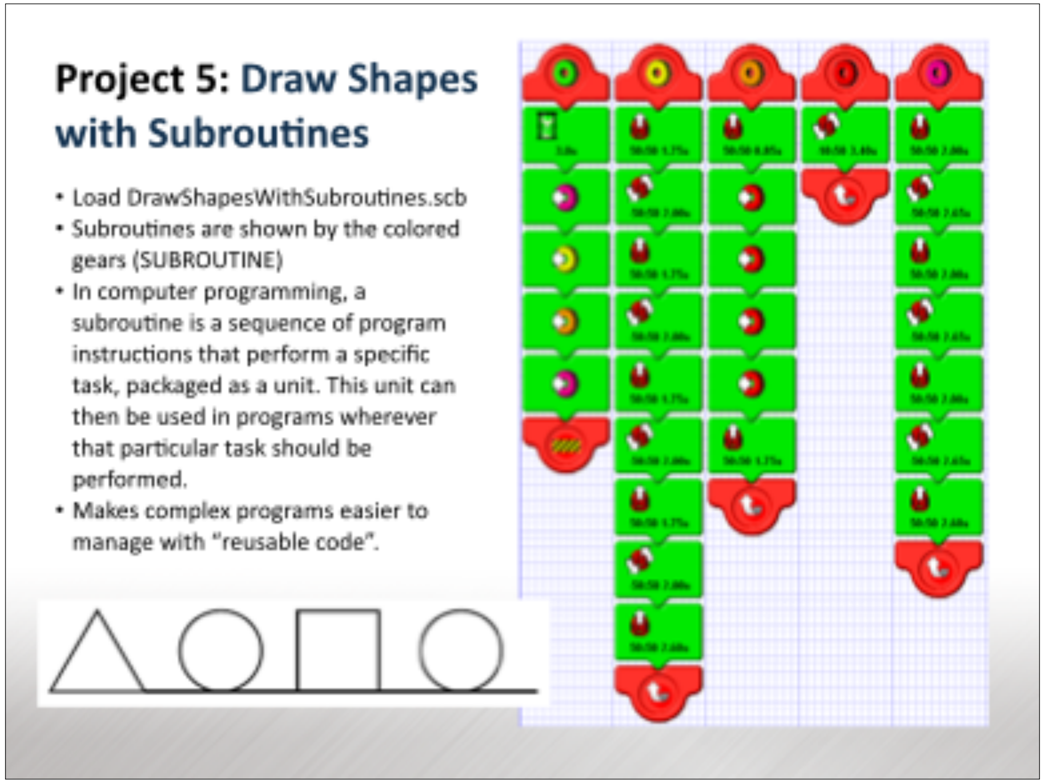
Discuss ways to increase time: LOOPs, providing 0 as a movement time
Program the S2 to create circles, squares and shapes
This is an example of open-loop robotic programming (no sensor feedback).
See also the Neil Rosenberg “Robots for Beginners” Project 4: A’maze”ing Turns
CCSS.MATH.CONTENT.6.NS.C.5 Understand that positive and negative numbers are used together to describe quantities having opposite directions or values (e.g., temperature above/below zero, elevation above/below sea level, credits/debits, positive/negative electric charge); use positive and negative numbers to represent quantities in real-world contexts, explaining the meaning of 0 in each situation.



DriveFigureEight.scb is an example of this program.

Precision Control: S2's Encoders

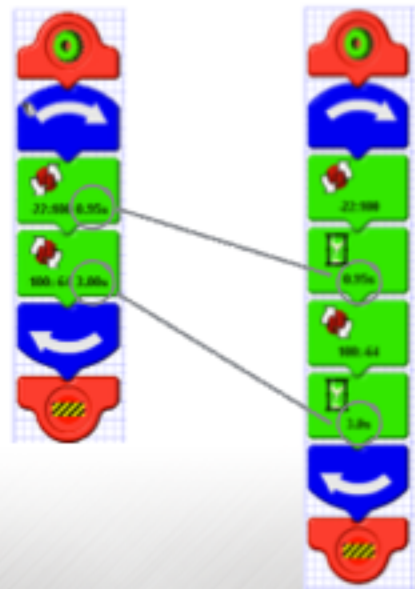
- 517 counts/ revolution
- Managed by the S2's Multicore Propeller P8X32A-Q44



See also the Neil Rosenberg "Robots for Beginners" Project 5: Subroutines and Project 6: Simple Figures, Loops.
CCSS.MATH.CONTENT.4.MD.C.6 Measure angles in whole-number degrees using a protractor. Sketch angles of specified measure.
CCSS.MATH.CONTENT.6.G.A.1 Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
CCSS.MATH.CONTENT.7.G.A.1 Solve problems involving scale drawings of geometric figures, including computing actual lengths and areas from a scale drawing and reproducing a scale drawing at a different scale.

Project 6: Spirographs

- Spirographs are easily-created repetitive patterns.
- You can build a spirograph with LOOP tiles or SUBROUTINES.
- Load SpiroSpiky.scb as an example.
- Note that instead of putting timing constants in the MOTION tile, you could also use zero (0) and add a PAUSE tile. Example is shown to the right.



Spirographs are limitless fun for students – you could easily spend several days on this aspect of the S2.

See also the Neil Rosenberg “Robots for Beginners” Competition 1: Drawing Trials.
Load the examples SpiroSpiky.scb, SpiroTriangleStar.scb and SpiroSmallFivePetalFlower.scb.

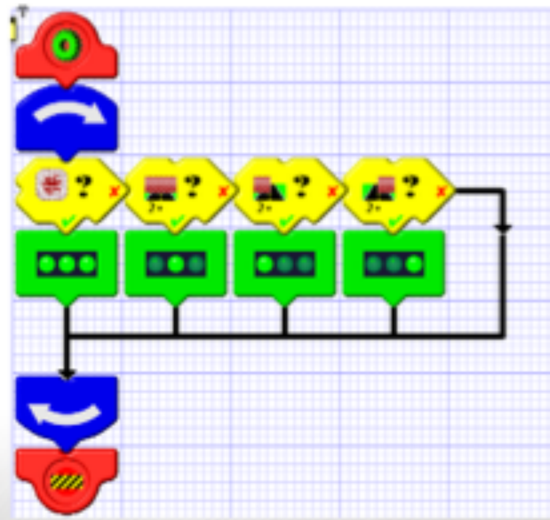
Project 7: Observation Deck

- Press “Observation Deck” button to download
- Sensor feedback from infrared emitters, light sensors, and line followers
- Provides a quick way to test all sensors and verify how they function within the robot’s environment

See also the Neil Rosenberg “Robots for Beginners” Project 8: The Monitor (no Merrimack?)

Project 8: Infrared Object Avoidance

- Load Observation Deck to see that infrared sensors are working properly
- Load InfraredSensing.scb and verify that everything functions correctly.

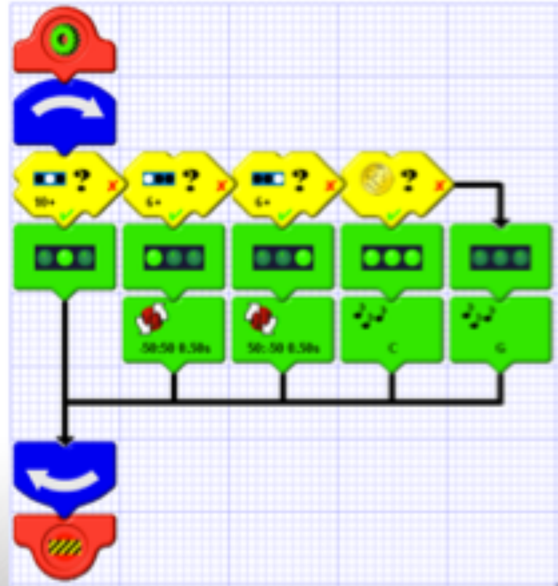


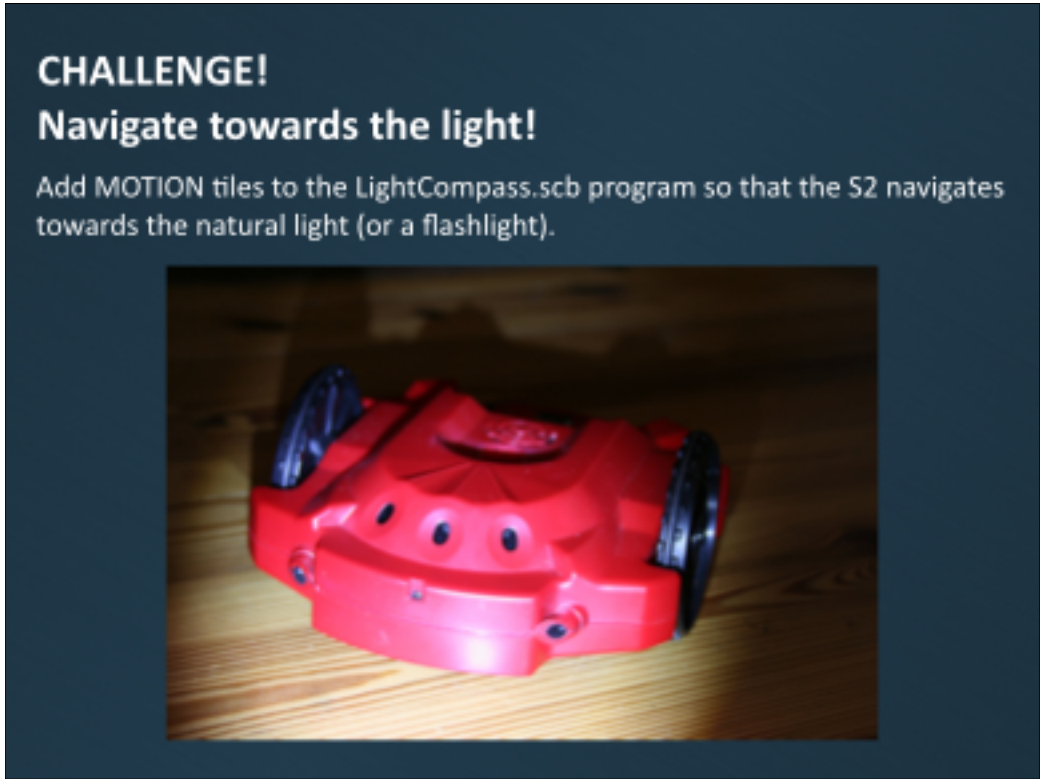
InfaredSensingMovement.scb provides a simple example of autonomous, roaming behavior.
Use FLAGs or RANDOM behavior to make this project more functional.
See also the Neil Rosenberg “Robots for Beginners” Project 13: Feeling your way

InfraredSensingMovement.scb provides an example to solving this challenge.
Obstacles could include boxes, books and other robots.

Project 9: Light Compass

- Run S2 Observation Deck
- Observe light level values
- Download LightCompass.scb
- Add and change MOTION tiles so the S2 follows the light

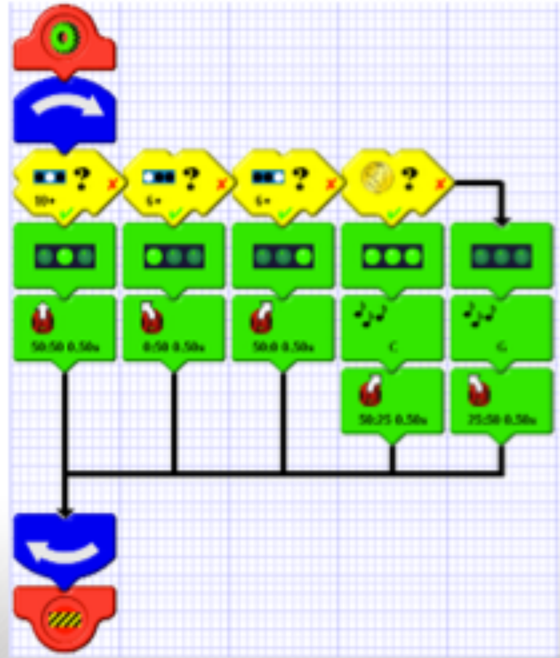




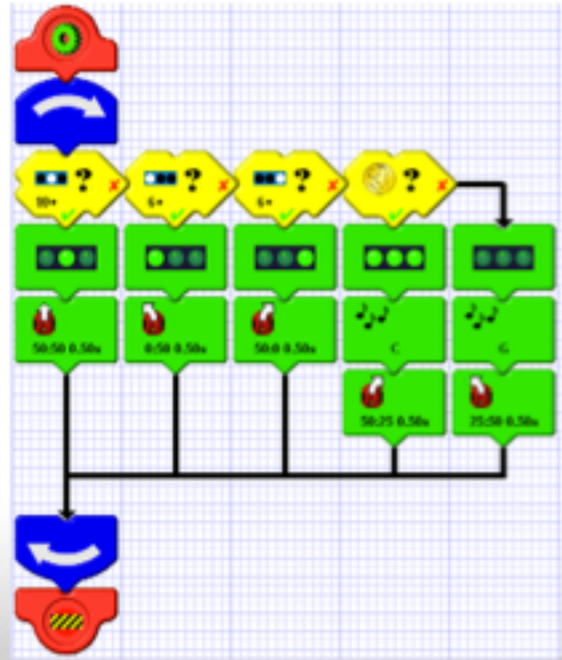
LightFollow.scb provides an example to solving this challenge.
What program additions could be made if the robot gets stuck? Additional challenge is to add infrared object navigation and subroutines into this program so that the S2 follows light and avoids objects on the way there.
The Neil Rosenberg Workshop – Robots for Beginners curriculum does not include a light following project.

Project 10: Light Following

- Open the Observation Deck and see the differences in light levels in your environment.
- Download LightFollowing.scb
- If there is no discernible difference in light levels, the RANDOM tile is used to move right or left.
- Could the program be improved?



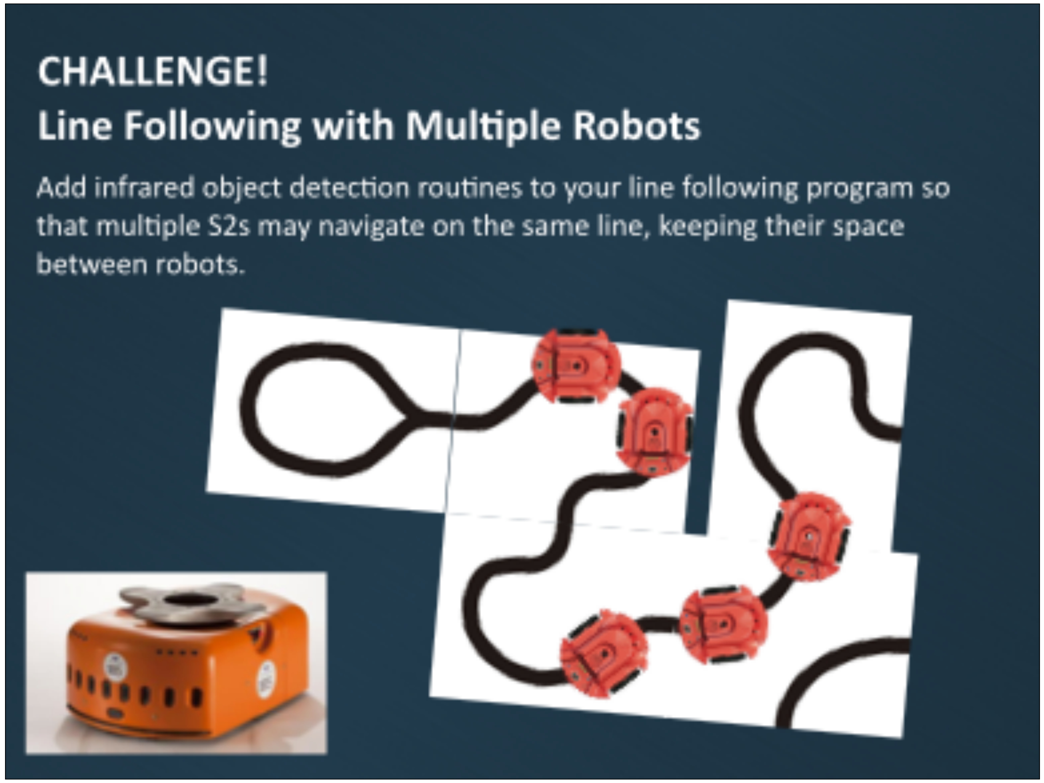
Project 11: Line Following



- Download LineFollowerLEDs.scb
- Test the program on a piece of black tape.

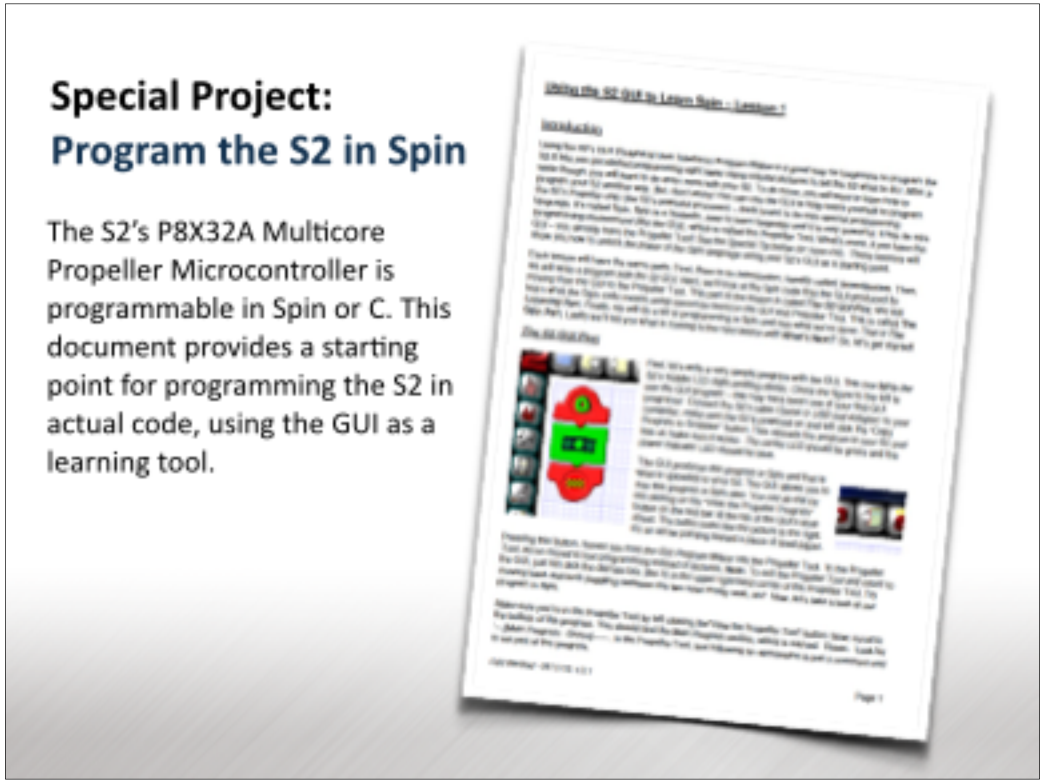
See also the Neil Rosenberg “Robots for Beginners” Project 9: Stop at the Line
See also the Neil Rosenberg “Robots for Beginners” Project 11: Follow the Black Brick Road (sorry Dorothy)
See also the Neil Rosenberg “Robots for Beginners” Project 12: Get Better at Following

LineFollowerDrive.scb provides an example to solving this challenge.
Line following works best with black masking tape. Electrical tape is often too reflective.

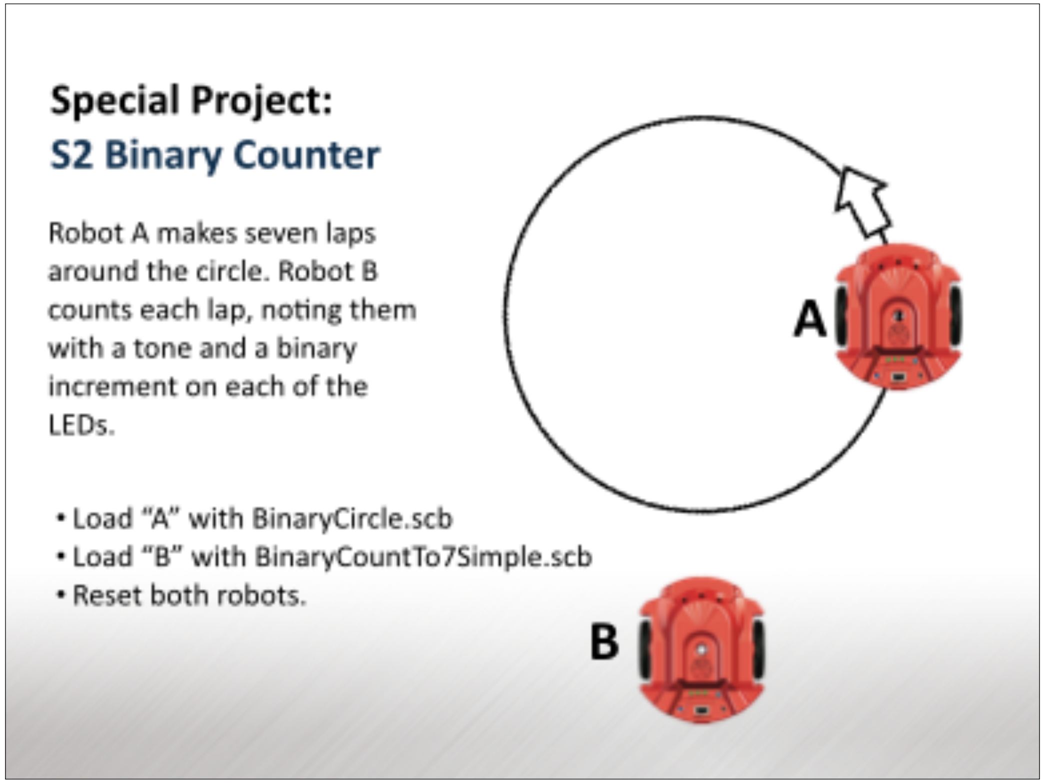


LightFollow.scb provides an example to solving this challenge.
The Neil Rosenberg Workshop – Robots for Beginners curriculum does not include a light following project.

LightFollow.scb provides an example to solving this challenge.
The Neil Rosenberg Workshop – Robots for Beginners curriculum does not include a light following project.



Thank you to Whit Stodghill for this contribution.

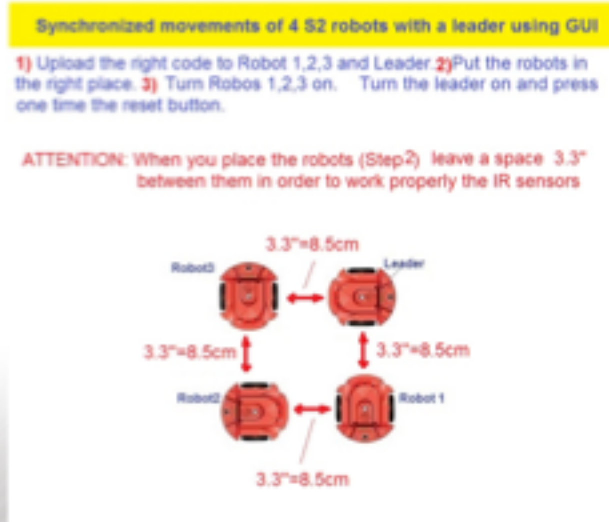


Thank you to Whit Stodghill for this contribution.

Special Project: Synchronize Four S2 Robots with a Leader

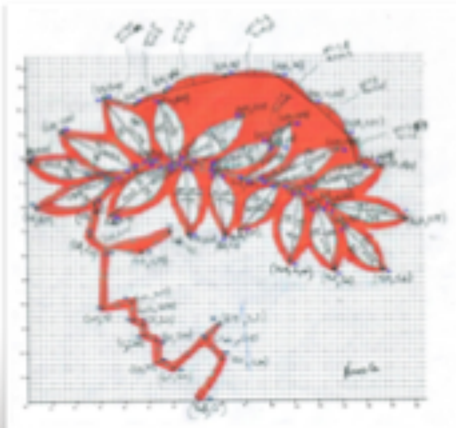
- S2 codes for this project are:
  - SynchroBot1.scb
  - SyncrhoBot2.scb
  - SynchroBot3.scb
  - SynchroLeader.scb

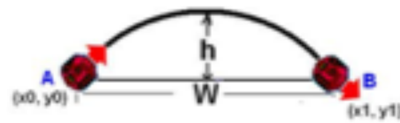
This project consists of a synchronized robot dance.



Special Project: Use Coordinates to Create Logos

- Entire project at http://forums.parallax.com/showthread.php/137549-Scribbling-sport-team-logos-with-S2-using-Microsoft-Excel





$$R = \frac{h}{2} + \frac{w^2}{8h}$$

$$w = \sqrt{(x1 - xo)^2 + (y0 - y1)^2}$$

$$R = \frac{h}{2} + \frac{(x1 - xo)^2 + (y0 - y1)^2}{8h}$$

Special Project: Messenger Bots Carry M&Ms

- Entire project at http://learn.parallax.com/s2-messenger-bots

S2 Messenger 'Bots

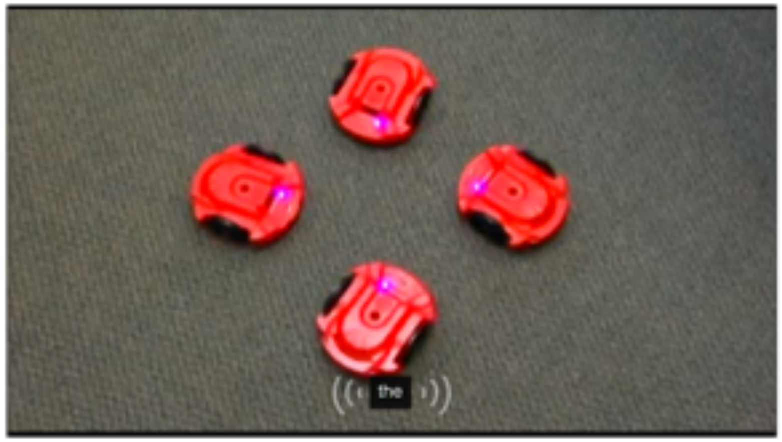
Submitted by Jessica Uelmen on Tue, 10/29/2013 - 14:12

Special Project: Hokey Pokey

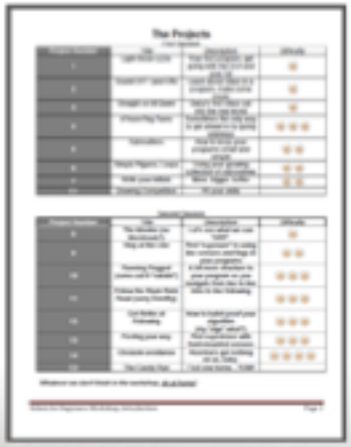
- Entire project at http://learn.parallax.com/s2-robot-hokey-pokey

The S2 Robot Hokey Pokey

Submitted by Jessica Uelmen on Fri, 10/04/2013 - 12:23



S2 Curriculum:
Neil Rosenberg's "Robotics for Beginners"



The Projects

- Download from www.parallax.com
- 36 pages, Teacher's Guide available
- 14 consecutive projects
- Freely available

Propeller P8X32A Applications:
New Zealand's Emirates America's Cup Boat

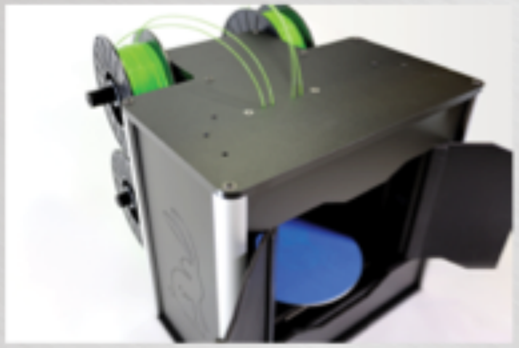
- 30 chips provide sensor feedback to captain

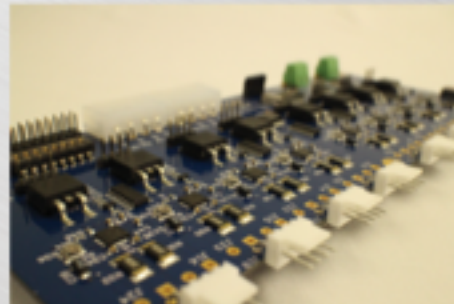
Propeller P8X32A Applications:
Parallax's ActivityBot Robot Kit

- Supported on learn.parallax.com with C programming tutorials

Propeller P8X32A Applications:
Radiant Fabrication's Lionhead 3D Printer

- 4 Propeller chips/printer for motor driving, SD card reading
- www.radiantfabrication.com





Propeller P8X32A Applications:

ELEV-8 Unmanned Aerial Vehicle

- Parallax's quadcopter is used in high school technology programs
- learn.parallax.com tutorials

S2 Curriculum:

Interactive Media's "Robotics for Beginners"

- From www.exploringrobotics.com
- 100 pages, Teacher's Guide available
- 20 consecutive projects
- 20 S2 Robots and Curriculum are $4,000

S2 Robot Purchasing Information

- Startup Guide
- Software
- Videos
- Line Following Tracks
- Neil Rosenberg Curriculum
- $129.00 each with 15% Educator discount

Running the Class:
Process, tips and fun

Laptops or computer lab
Space requirements
Running a class
Robo-Friday: soldering, show-and-tell, advanced robotics
STEM concepts in robotics
Competitions in the one robot per student environment

PARALLAX INC

www.parallax.com