

S2 Robot Educator's Course



Ken Gracey, CEO
Parallax Inc.

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



S2 Educator's Course Agenda

- A. 8:30 to 9:00 Welcome and Introductions
- B. 9:00 to 9:30 S2 Graphical User Interface
- C. 9:30 to 10:30 Lights, Sound and Motion
- D. 10:30 to 10:45 Break
- E. 10:45 to 12:00 Drawing Shapes and Spirographs with Sharpie
- F. 12:00 to 1:00 Lunch Break
- G. 1:00 to 2:30 Navigation using Sensors
- H. 2:30 to 3:00 Robotics in Class - Teaching
 - 1. S2 Curriculum Choices
 - 2. Applying Common Core Standards and Robotics
 - 3. Next steps in robotics for middle school and high school
 - 4. More S2 Projects
 - 5. Commercial applications of the Propeller



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.

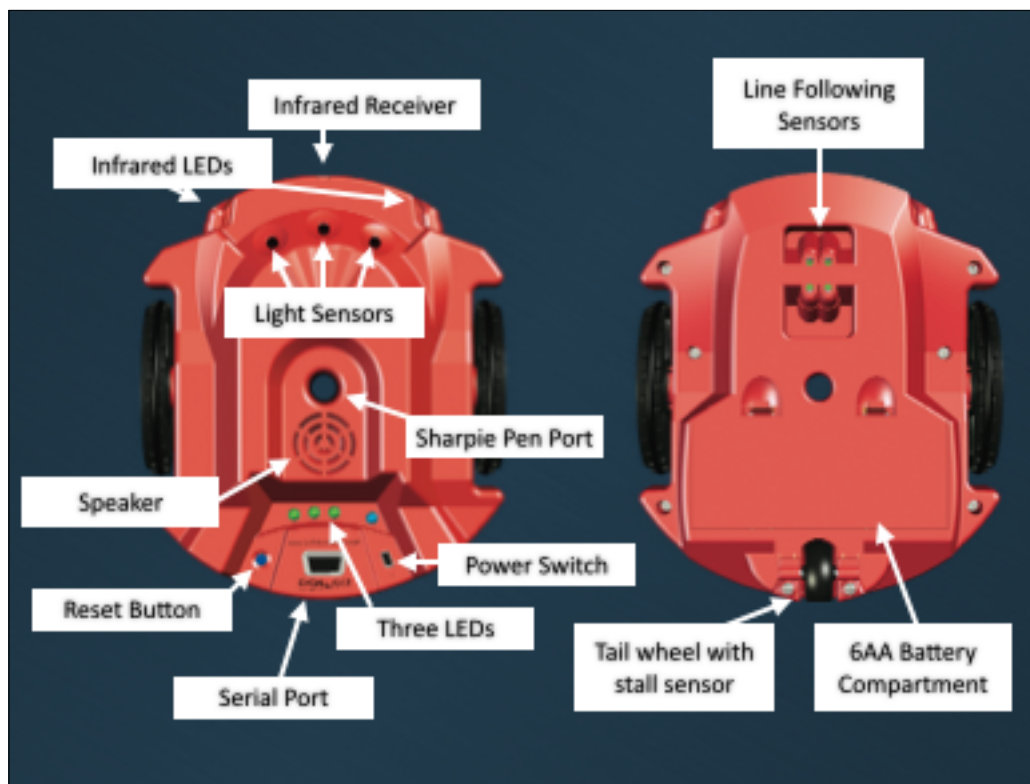
The most successful careers happen when personal and professional interests collide!



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.

Project 1: Light the LEDs

- Load LEDsOnOff.scb
- Change LOOP counter and PAUSE values.



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.

CHALLENGE!

Use your LEDs to make a cylon light chaser!

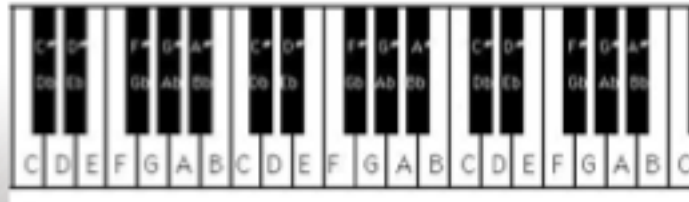
Modify the previous program LEDsOnOff.scb to be a cylon light chaser.



LEDsCylonChaser.scb provides an example to solving this challenge.

Project 2: Sounds

- Load Sounds.scb
- Experiment with notes, trumpet tunes and sound effects
- Consider using these to “debug” your programs



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.

Project 3: Movement

- Load DriveTwoSpeeds.scb
- Compare the duration and velocity of the two movements in this program
- Familiarize with velocity, timing, and physical results of the command
- $\text{distance} = \text{rate} \times \text{time}$



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.

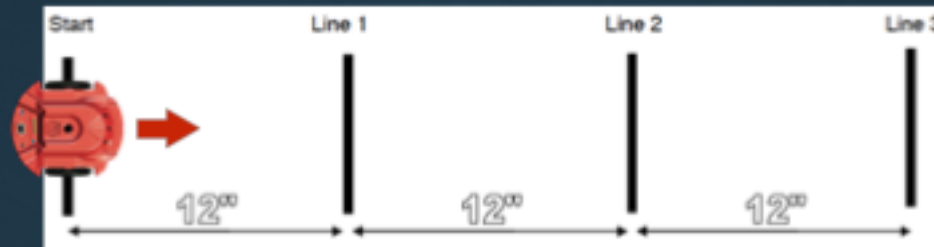
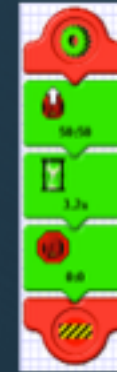
CCSS.MATH.CONTENT.4.MD.C.6 Measure angles in whole-number degrees using a protractor. Sketch angles of specified measure.

CHALLENGE!

Distance = rate x time for “dead reckoning”

At a speed of 50, 50 the robot travels 12” in 3.2s.

1. Open DriveTwelveInches.scb (shown on right).
2. Calculate the time required to travel 36”.
3. Modify the program so the S2 drives 36” forward, reverses and returns to the starting point.
4. With a MOTION tile speed of 75, how are the PAUSE values changed?



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

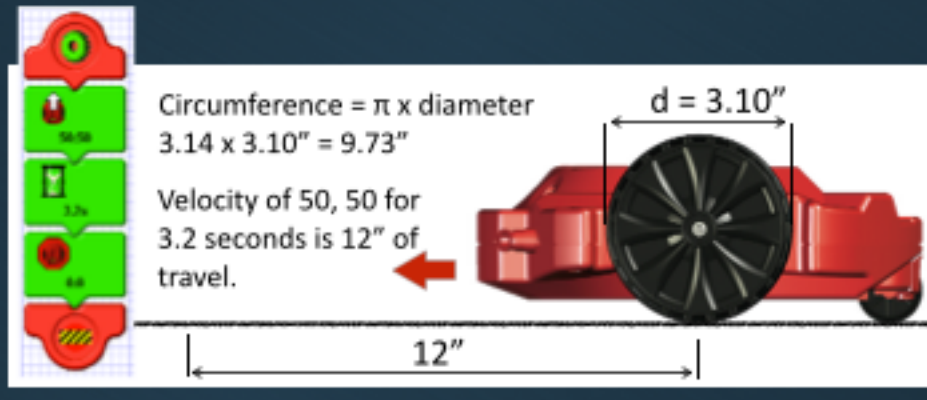
“Using Robotics to Promote Learning in Early Grades” http://www.asee.org%2Fpublic%2Fconferences%2F8%2Fpapers%2F5480%2Fdownload&ei=MLY-VJ78EcWZ8gHvxIGoDg&usq=AFOjCNGcHZJ_tCoB1-utqnCH1F2876wajg&sig2=-tm1L7i3c6bZCIXsHuRfHg&bvm=by.77412846.d.b2U

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.

CHALLENGE!

Circumference, time and distance travelled

1. At a speed of 50, 50 how many seconds of travel are required for a full rotation?
2. How many seconds are required to travel 18" at a speed of 50, 50?
3. Put a Sharpie pen into the S2's pen holder and verify with an S2.



Circumference = $\pi \times \text{diameter}$
 $3.14 \times 3.10'' = 9.73''$

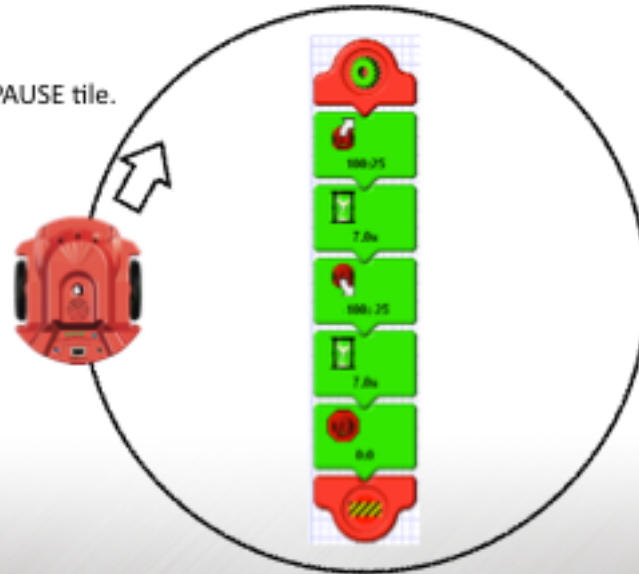
Velocity of 50, 50 for 3.2 seconds is 12" of travel.

Divide the circumference into 12" to obtain seconds, or $9.73'' / 12'' = 0.81$. Multiply by 3.20 seconds = $0.81 \times 3.20s = 2.60s$.

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.

Project 4: Circles

- Load DriveCircles.scb
- Time managed with PAUSE tile.



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).

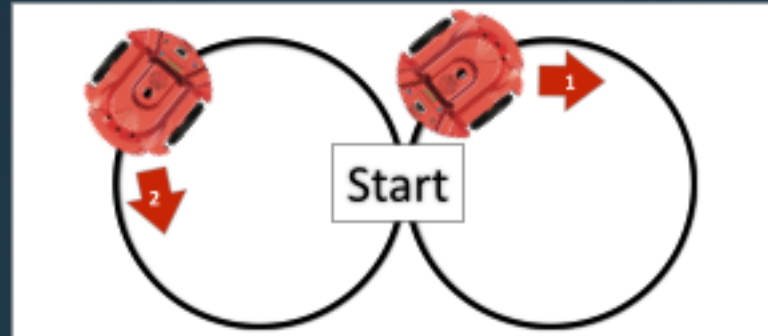
See also the Neil Rosenberg "Robots for Beginners" Project 4: A'maze'ing Turns

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.

CHALLENGE!

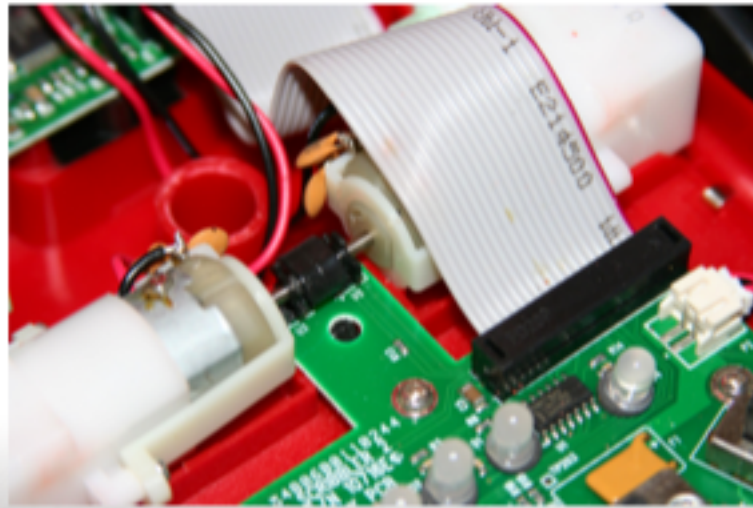
Draw a figure eight

1. Write an S2 program to draw a figure eight, starting in the middle.
2. Use the PAUSE tile to decide how long to drive.
3. Sharpies are optional to verify your start and finish location.



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



Project 5: Draw Shapes with Subroutines

- Load DrawShapesWithSubroutines.scb
- Subroutines are shown by the colored gears (SUBROUTINE)
- In computer programming, a subroutine is a sequence of program instructions that perform a specific task, packaged as a unit. This unit can then be used in programs wherever that particular task should be performed.
- Makes complex programs easier to manage with "reusable code".



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 spiograph 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.

CHALLENGE!

Create your own spirographs

Write your own S2 program with a few MOTION tiles in a loop. Work from one of the examples below or create your own. Use LOOPS and SUBROUTINES.



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

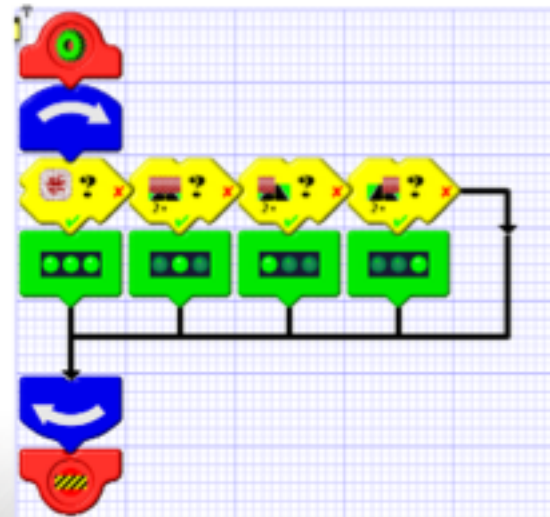
- Press “Observation Deck” button to download
- Sensor feedback from infrared emitters, light sensors, and line followers
- Provides a quick way to test all sensors and verify how they function within the robot’s environment



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.



InfraredSensingMovement.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

CHALLENGE!

Obstacle avoidance using infrared and stall sensor.

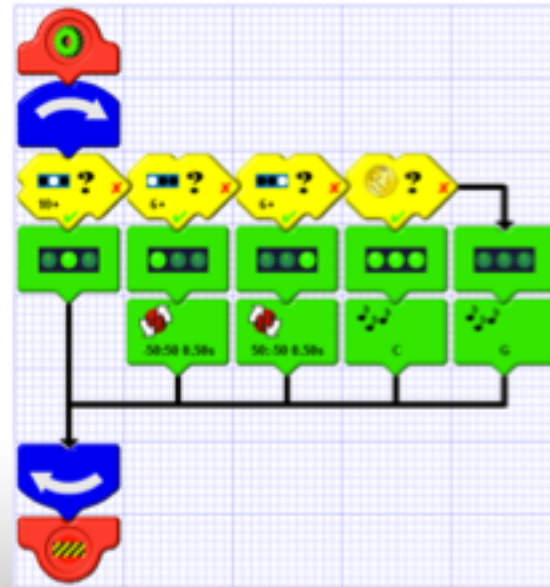
Add MOTION tiles to your S2's InfraredSensing.scb program so your robot may roam autonomously using both the stall detection and infrared sensors.



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



CHALLENGE!

Navigate towards the light!

Add MOTION tiles to the LightCompass.scb program so that the S2 navigates towards the natural light (or a flashlight).



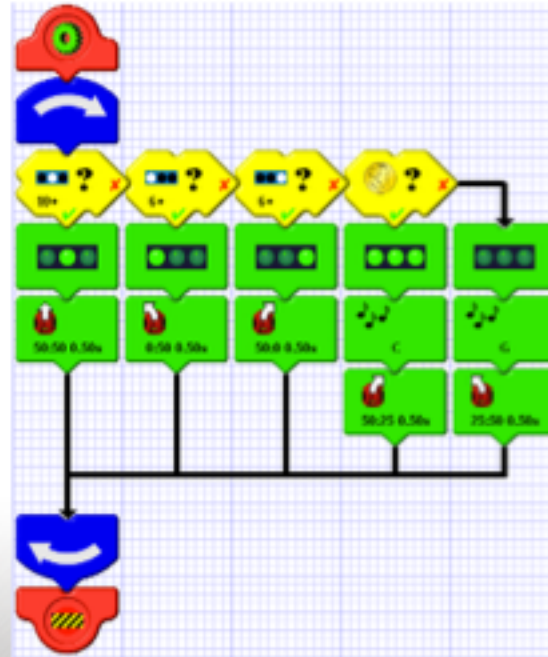
LightFollow.scb provides an example to solving this challenge.

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.

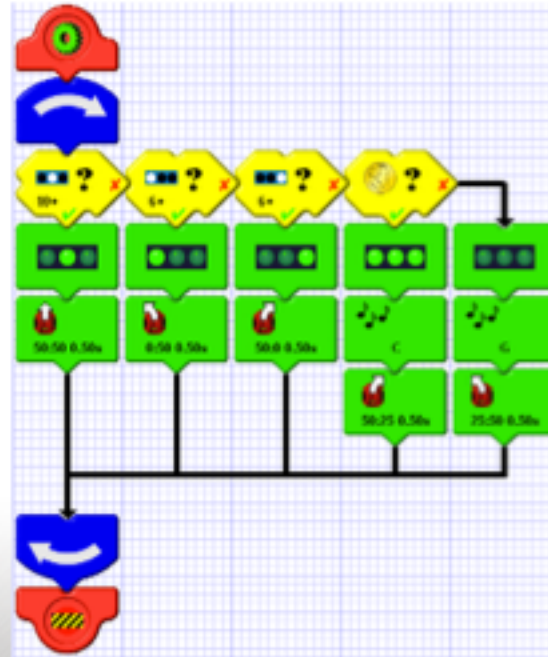
Project 10: Light Following

- 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

CHALLENGE!

Make your S2 robot follow a black line

Add MOTION tiles to the LineFollowerLEDs.scb program so that the S2 follows a line. Can you program the S2 to navigate across line breaks?

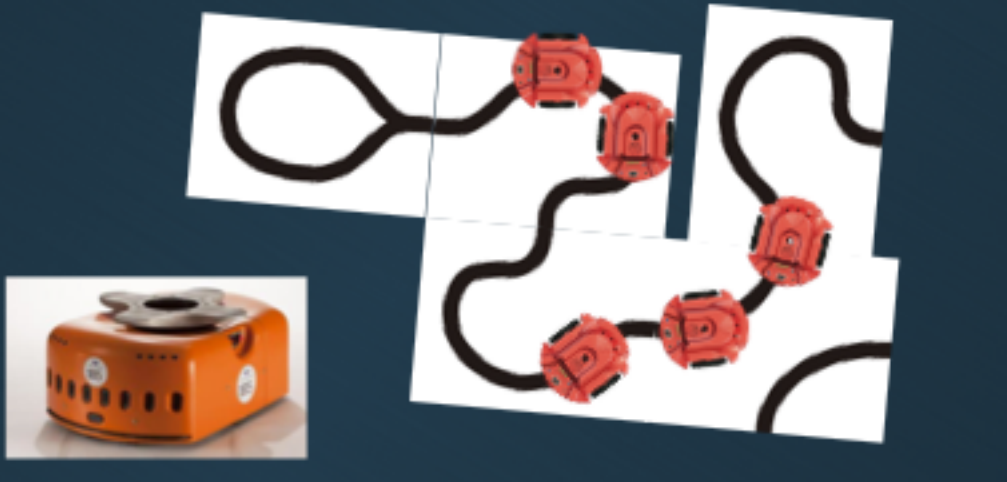


LineFollowerDrive.scb provides an example to solving this challenge.
Line following works best with black masking tape. Electrical tape is often too reflective.

CHALLENGE!

Line Following with Multiple Robots

Add infrared object detection routines to your line following program so that multiple S2s may navigate on the same line, keeping their space between robots.



LightFollow.scb provides an example to solving this challenge.

The Neil Rosenberg Workshop – Robots for Beginners curriculum does not include a light following project.



S3 Robot Brief Overview

- A. Lithium-ion rechargeable battery pack.
- B. USB programming port.
- C. Wireless communication and programming socket.
- D. Fully open-sourced, design progress on Parallax forums.
- E. Expected delivery 2016.

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

Special Project: Program the S2 in Spin

The S2's P8X32A Multicore Propeller Microcontroller is programmable in Spin or C. This document provides a starting point for programming the S2 in actual code, using the GUI as a learning tool.

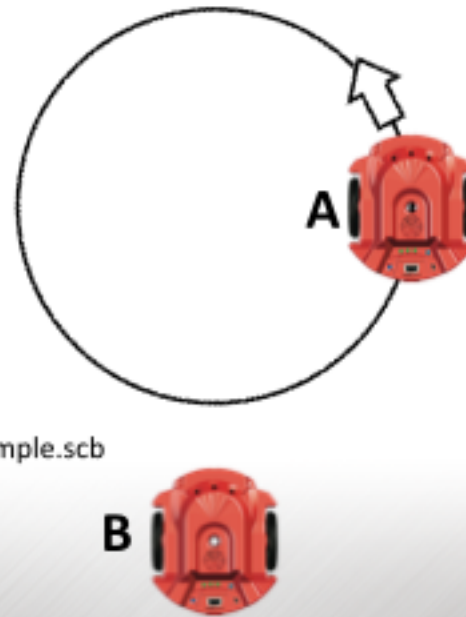


Thank you to Whit Stodghill for this contribution.

Special Project: S2 Binary Counter

Robot A makes seven laps around the circle. Robot B counts each lap, noting them with a tone and a binary increment on each of the LEDs.

- Load "A" with BinaryCircle.scb
- Load "B" with BinaryCountTo7Simple.scb
- Reset both robots.



Thank you to Whit Stodghill for this contribution.

Special Project: Synchronize Four S2 Robots with a Leader

• S2 codes for this project are:

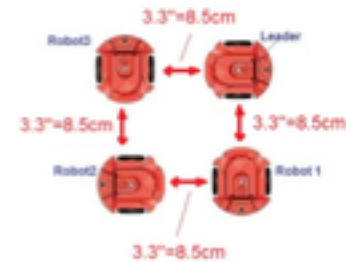
SynchroBot1.scb
SynchrhoBot2.scb
SynchroBot3.scb
SynchroLeader.scb

Synchronized movements of 4 S2 robots with a leader using GUI

1) Upload the right code to Robot 1,2,3 and Leader. 2) Put the robots in the right place. 3) Turn Robos 1,2,3 on. Turn the leader on and press one time the reset button.

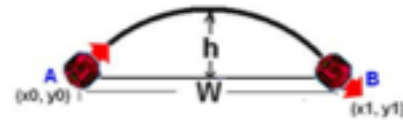
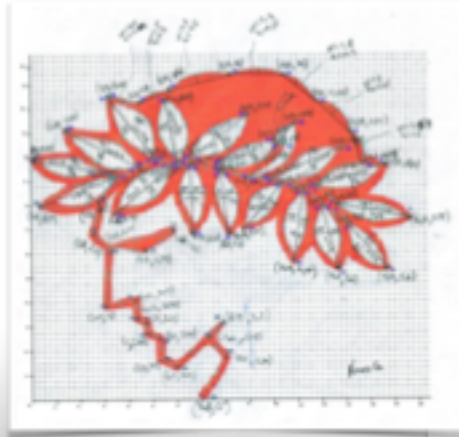
ATTENTION: When you place the robots (Step 2) leave a space $3.3''=8.5\text{cm}$ between them in order to work properly the IR sensors

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>



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

$$w = \sqrt{(x1 - x0)^2 + (y0 - y1)^2}$$

$$R = \frac{h}{2} + \frac{(x1 - x0)^2 + (y0 - y1)^2}{8h}$$

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

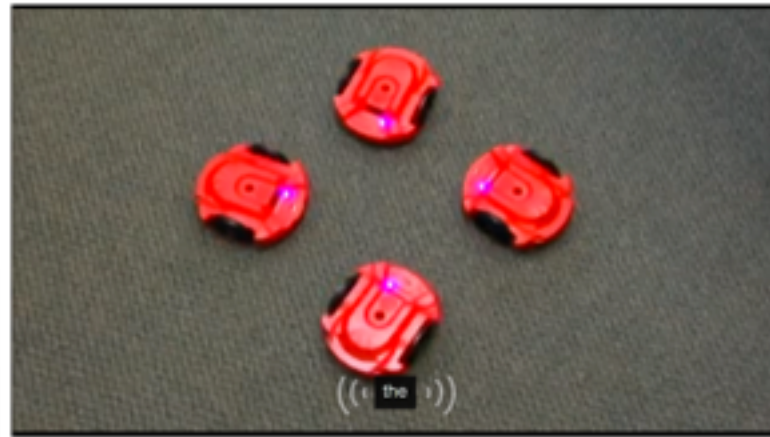


Special Project: Hokey Pokey

- 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			
Project	Topic	Difficulty	Stars
1	Light Sensor	Very Easy	★
2	Temperature Sensor	Very Easy	★
3	Sound Sensor	Very Easy	★
4	Ultrasonic Sensor	Very Easy	★
5	Line Follower	Easy	★★
6	Robot Navigation	Easy	★★
7	Line Follower	Easy	★★
8	Line Follower	Easy	★★
9	Line Follower	Easy	★★
10	Line Follower	Easy	★★
11	Line Follower	Easy	★★
12	Line Follower	Easy	★★
13	Line Follower	Easy	★★
14	Line Follower	Easy	★★

- 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



How it Works

The ActivityBot robot kit is a complete, ready-to-use robot that can be programmed with the Propeller P8X32A microcontroller. The robot is built around a metal chassis and features two large wheels and a smaller front wheel. The Propeller P8X32A microcontroller is connected to the robot's sensors and actuators, allowing it to perform a variety of tasks.

Key features of the ActivityBot robot kit include:

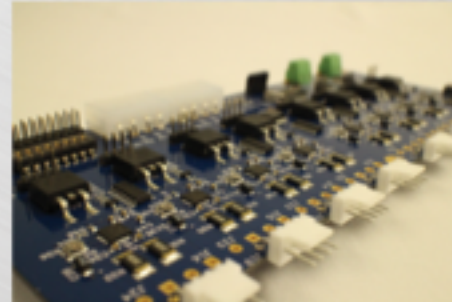
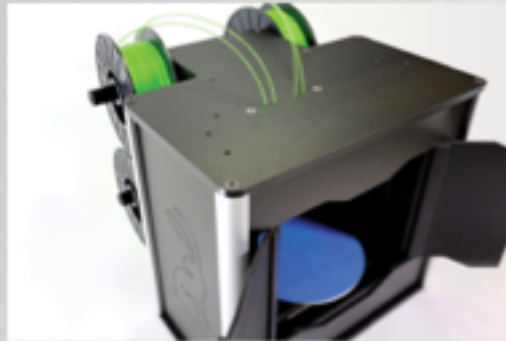
- Propeller P8X32A microcontroller
- Propeller P8X32A microcontroller
- Propeller P8X32A microcontroller

```
1 // Propeller P8X32A microcontroller
2 // Propeller P8X32A microcontroller
3 // Propeller P8X32A microcontroller
4 // Propeller P8X32A microcontroller
5 // Propeller P8X32A microcontroller
6 // Propeller P8X32A microcontroller
7 // Propeller P8X32A microcontroller
8 // Propeller P8X32A microcontroller
9 // Propeller P8X32A microcontroller
10 // Propeller P8X32A microcontroller
11 // Propeller P8X32A microcontroller
12 // Propeller P8X32A microcontroller
13 // Propeller P8X32A microcontroller
14 // Propeller P8X32A microcontroller
15 // Propeller P8X32A microcontroller
16 // Propeller P8X32A microcontroller
17 // Propeller P8X32A microcontroller
18 // Propeller P8X32A microcontroller
19 // Propeller P8X32A microcontroller
20 // Propeller P8X32A microcontroller
21 // Propeller P8X32A microcontroller
22 // Propeller P8X32A microcontroller
23 // Propeller P8X32A microcontroller
24 // Propeller P8X32A microcontroller
25 // Propeller P8X32A microcontroller
26 // Propeller P8X32A microcontroller
27 // Propeller P8X32A microcontroller
28 // Propeller P8X32A microcontroller
29 // Propeller P8X32A microcontroller
30 // Propeller P8X32A microcontroller
31 // Propeller P8X32A microcontroller
32 // Propeller P8X32A microcontroller
33 // Propeller P8X32A microcontroller
34 // Propeller P8X32A microcontroller
35 // Propeller P8X32A microcontroller
36 // Propeller P8X32A microcontroller
37 // Propeller P8X32A microcontroller
38 // Propeller P8X32A microcontroller
39 // Propeller P8X32A microcontroller
40 // Propeller P8X32A microcontroller
41 // Propeller P8X32A microcontroller
42 // Propeller P8X32A microcontroller
43 // Propeller P8X32A microcontroller
44 // Propeller P8X32A microcontroller
45 // Propeller P8X32A microcontroller
46 // Propeller P8X32A microcontroller
47 // Propeller P8X32A microcontroller
48 // Propeller P8X32A microcontroller
49 // Propeller P8X32A microcontroller
50 // Propeller P8X32A microcontroller
```



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



The screenshot shows the Parallax website interface. At the top, there is a navigation menu with links for HOME, PRODUCTS, TUTORIALS, SUPPORT, and CONTACT. Below the navigation, there is a search bar and a list of tutorial categories. The main content area displays a tutorial titled "Tutorial 1: Building the Flight Controller and Motor". The tutorial text describes the process of building a flight controller and motor, including the required components and the steps to follow. A small image of the drone is also visible in the bottom right corner of the screenshot.

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

