Web Site: www.parallax.com
Learn: learn.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267
EDU Hotline: (916) 625-6801

Spin Project: Christmas Scribbler

Level: Intermediate
Skills Required: Soldering, spin programming, S2 GUI programming
Hours to Complete: 2

View this project's YouTube video on the ParallaxInc YouTube channel.

Since the S2 Robot is already red, it really lends itself well to holiday projects. This project uses the S2 Robot to play 'O Tannenbaum' to objects it sees. Using the S2 Robot's hacker port, lights on a hacked Christmas tree blink in time with the song.

Small Christmas trees like the one used in this project can be found at almost any craft store (this one was actually nabbed from Home Depot). Most come with a continuous strand of LEDs wound around the tree, so the strand will need to be cut at several points in order to create an alternating blinking effect. Since the cut wires aren't easily connected to the S2's hacker port, pluggable female wires will need to be soldered to the LED wires.

Programming the S2 Robot will be accomplished using a combination of the S2 GUI and the Propeller Tool. The S2 GUI will be used to program the music and navigation code, but the software doesn't allow direct access to the Propeller I/O connected to the S2 Hacker Port. That's when the Propeller Tool to will be used to add code to control the Christmas tree's LEDs.

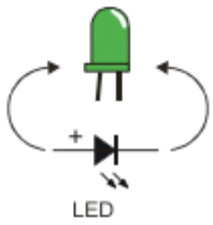
What's Needed

- (1) S2 Robot
- (1) Small lighted Christmas Tree
- Soldering Iron
- Solder
- Wire Cutters
- Wire Strippers
- Heat shrink tube or electrical tape
- Additional wire
- Download the S2 GUI

Hack the S2 and Tree

In order to create an alternating light sequence on the Christmas tree, the string of LEDs need to be cut at various points. Keep in mind that the ground connection can remain common across the string, so you can decrease the number of wires that need to be soldered.

Did you know?
LEDs have polarity, meaning there is a positive and negative connection. When hacking the Christmas tree, be sure to only cut the positive wires since these will connect to the Propeller I/O pins through the S2 hacker port.



Most Christmas tree LEDs look like the one shown above. The flat spot on the LED package designates the negative connection, so be sure to not cut that wire.

- ✓ Cut the LED string on the tree in a few places, keeping the ground wire intact.
- ✓ After cutting the wires, strip the cut ends so that about 0.25” of wire is exposed.

✓ Cut the pluggable female wire to the desired length and strip the same amount.

✓ Before soldering, twist the two wires together to create a mechanical bond.

- ✓ Solder the wires together, using the soldering iron to heat the wire from the bottom and add solder to the top. Then, insulate the connection with heat shrink tubing or electrical tape.
- ✓ Open your S2 by removing the six screws on the bottom of the robot.
- ✓ Connect the Christmas tree wires to S2 Hacker port. For a cleaner display, thread the wires through the S2 pen port to hide them. Connect the positive LED wires to pins 2-5 on the Propeller, and your ground wire to one of the ground pins.

- ✓ Put your S2 Robot back together and attach the tree to the top of the case using Velcro or double-sided sticky tape.

Go Caroling

- ✓ Open the code included in the project download.
- ✓ Load the program to your S2 by pressing F11 or selecting Run -> Compile Current -> Load EEPROM.

Once loaded, your robot should travel in a straight line until it finds an object directly in its path. Once the robot finds an object, it will excitedly play ‘O Tannnenbaum’ while blinking lights on the Christmas tree. When the song has ended, the robot with back up, turn and continue travelling in a straight line until it finds a new object to play the song to.

How Mixing GUI and Spin Code Works

The S2 GUI software provides an easy way to program the S2 using graphical tiles. However, there are some limitations that come with the software. For example, you cannot directly control the I/O pins connected to the S2 Hacker Port using the GUI software.

In these cases, we need to use the programming language native to the S2 Robot's Propeller microcontroller, Spin. A nice feature of the S2 GUI software is that you can easily translate the GUI worksheet to Spin code and add to it using the Propeller Tool.

In the case of the S2 Caroling Robot, the navigation and music code was written using the S2 GUI Software. When done, the worksheet looks like so:

Did you know? The Help Section of the S2 GUI software provides a wealth of information on the function and capabilities of each worksheet tile. You can access it by clicking the question mark button at the top of the screen.

While this worksheet covers the majority of the code for the caroling robot, we need to modify the Spin program in order to control the LEDs on the hacked Christmas tree.

- ✓ Do so by clicking the ‘View the Propeller program’ button in the top navigation bar.

This opens the Propeller Tool. The code translated from the S2 worksheet is found in the PUB Green and PUB Yellow sections, which correspond to the gear colors in the GUI worksheet.

The `PUB` sections are known as *methods* in Spin. We won't go into the specifics of the Spin programming language here, but if you want to learn more there are great tutorials available on learn.parallax.com/PropellerBOE.

Adding the following code under the `PUB Yellow` method will blink the LED strands, alternating on and off between the strands connected to pins 5 & 3 and to pins 4 & 2.

The `Blink` method will then need to be called from our main program. We'll want the LEDs to blink when the music plays so the code will be added after the robot is stopped, but before the music begins.

Since the Propeller is a multicore microcontroller, executing multiple tasks simultaneously is a breeze. The command `cognew` launches a new processor (or cog as they're referred to in Spin), where `Blink` calls the `Blink` method and `@ledstack` sets aside program space for the processor to run.

`@ledstack` is then a variable declared in the `VAR` block at the top of the program.

You can follow this same process to customize the tune in your own Caroling Scribbler.