
RSeries Lights Module Documentation

Release latest

April 26, 2016

1 Where To Buy? 3
1.1 Versions 3

Logic Displays are the blinky lights in the rectangular openings of R2-D2's dome (2 front, 1 rear) that separate dead droids from the living.

The RSeries Logic Engine combines Arduino, RGB LEDs and fiber-optics to accurately emulate the color-wheel logic display patterns seen in the original Star Wars trilogy. This is arguably the most screen-accurate logic display system since Return of The Jedi.

At it's heart is an AVR Board (with Arduino Uno bootloader) which runs a series of WS2812B LEDs. These RGB LEDs cross-fade between set colors. For example the Front logic LEDs fade from black to blue to white and back again. As it arrives at each "key" color, that LED is paused for a random length of time - this creates constantly evolving color patterns that appear almost random at first glance.

Fiber-optic cable is used to route the light from each LED to specific 'pixels' on the logic bezel, which is what is finally seen on your droid.

Where To Buy?

The Logic Engine is supplied in DIY kits that are prepared exclusively for members of the R2 Builders Club. Look in [the ‘Other Parts’ forum on Astromech.net](<http://astromech.net/forums/forumdisplay.php?63>) to check current availability.

1.1 Versions

Read the Docs supports multiple versions of your repository. On the initial import, we will create a `latest` version. This will point at the default branch for your VCS control: `master`, `default`, or `trunk`.

We also create a `stable` version, if your project has any tagged releases. `stable` will be automatically kept up to date to point at your highest version.

1.1.1 How we envision versions working

In the normal case, the `latest` version will always point to the most up to date development code. If you develop on a branch that is different than the default for your VCS, you should set the **Default Branch** to that branch.

You should push a **tag** for each version of your project. These tags should be numbered in a way that is consistent with [semantic versioning](#). This will map to your `stable` branch by default.

Note: We in fact are parsing your tag names against the rules given by [PEP 440](#). This spec allows “normal” version numbers like `1.4.2` as well as pre-releases. An alpha version or a release candidate are examples of pre-releases and they look like this: `2.0a1`.

We only consider non pre-releases for the `stable` version of your documentation.

If you have documentation changes on a **long-lived branch**, you can build those too. This will allow you to see how the new docs will be built in this branch of the code. Generally you won’t have more than 1 active branch over a long period of time. The main exception here would be **release branches**, which are branches that are maintained over time for a specific release number.

1.1.2 Redirects on root URLs

When a user hits the root URL for your documentation, for example `http://pip.readthedocs.org/`, they will be redirected to the **Default version**. This defaults to **latest**, but could also point to your latest released version.