Contributing to Esp-Link

========================

Esp-link is not the work of a single individual, rather many people have contributed directly or indirectly.

Your contribution is very much appreciated, but please follow these guidelines to make the task easier on

everyone.

- Contributions do not have to be in the form of code: often documentation, how-tos are very valuable and answering questions

in github issues as well as gitter is also very valuable and welcome!

- Before you make a change or submit a change via a pull request, \*\*open an issue and discuss your proposed change\*\*. Gitter

is a good alternative to a github issue. This ensures that you don't spend time doing work that ultimately won't be accepted.

There's nothing more frustrating than receiving a pull-request that has lots of goodies but doesn't fit because it wasn't

discussed and agreed upon up-front.

- Keep your pull request as small as practical, if you have 3 things you want to change, please create 3 pull requests,

or at the very least, make sure your 3 changes are in different commits. This makes the review and testing easier

and ensures that if one feature is good to go it can be merged even if another feature needs more tweaking.

- The esp-link codebase is not uniform, it comes from a variety of sources, in particular esphttpd. A result of this is

that there is more than one coding style in use. If you make changes to existing files, please respect the file's

coding style (yes, sometimes that's not even totally uniform). Your overall goal should be for your code or changes to

look as if the original author had made them, not how you would like them to look.

- Changes that reformat or reorganize code will generally not be accepted, please do not mix them with other functionality

changes you are making and certainly discuss them first. Accept the fact that some people prefer bastards over pure-breads ;-).

- Esp-link has a mission stated in the readme.md, changes that deviate from that mission will generally be rejected. The reason

is that at the end of the day focusing on doing one thing well has a higher chance of succeeding than doing many things.

In that sense, esp-link is not a swiss-army knife firmware. (This being said, many people have used esp-link as a basis to add

their own functionality, which is very cool.)

I believe the above guidelines are pretty standard across a very large number of open source projects and not unique to esp-link,

so please do not get discouraged. Thank you for taking a look at esp-link!