Playing with the lights

Control WiFi-enabled LIFX light bulbs

Louis Opter <louis@opter.org>

Fosdem 2017, IoT track

$ whoami

Hello, my name is Louis (Opter) and I:

- am a decent software engineer;
- *do not really know anything about hardware.*

Anyway, it doesn’t really matter, let’s get started!

Agenda

Two related projects to talk about:

monolight	An UI for a 128 buttons matrix and lightsd;
lightsd	A daemon to easily control LIFX light bulbs.

Agenda

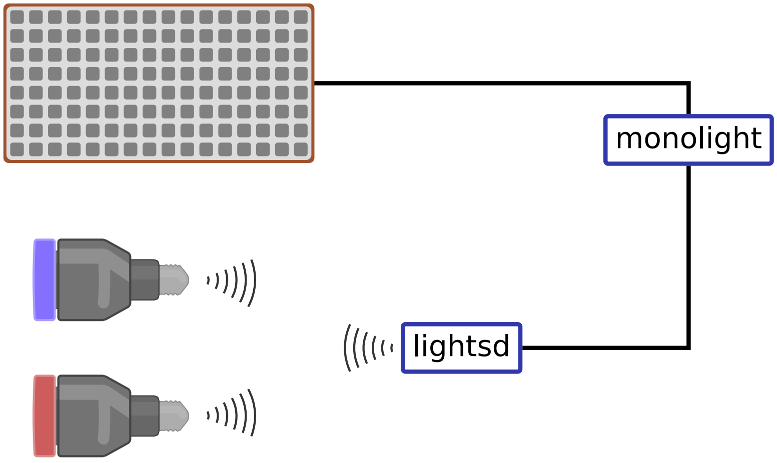
Two related projects to talk about:

monolight	An UI for a 128 buttons matrix and lightsd;
lightsd	A daemon to easily control LIFX light bulbs.

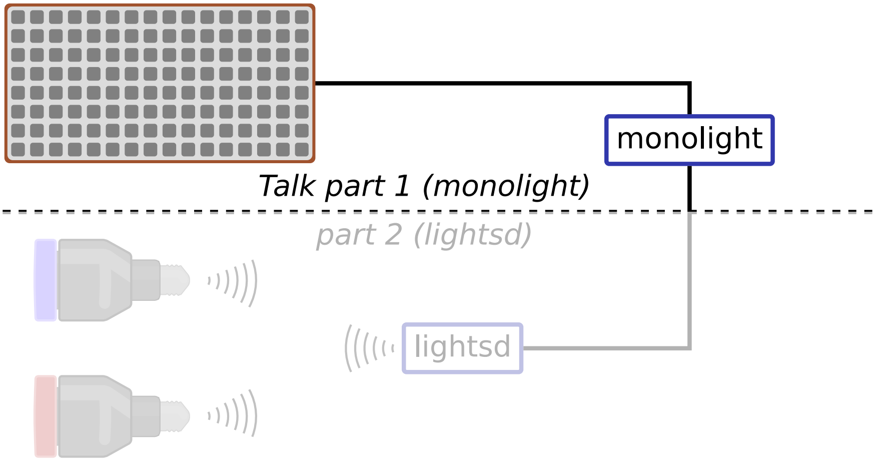
Outline:

- monolight: explanation, demo, implementation, ideas;
- lightsd: API demo, implementation, ideas, about LIFX;
- Q&A, discussion.

High-level architecture



High-level architecture



monolight

A controller (Monome grid 128 varibright):

- A matrix of 128 programmable button;
- 16 levels of brightness per button;
- Serial/RS232 (FTDI) connection.

monolight

A controller (Monome grid 128 varibright):

- A matrix of 128 programmable button;
- 16 levels of brightness per button;
- Serial/RS232 (FTDI) connection.

Controlling a "smart" bulb (LIFX Original 1000):

- A ~1000 lumens programmable light bulb;
- Nice colors, nice range of whites;
- Wi-Fi 802.11bgn, 2.4gHz.

monolight

A controller (Monome grid 128 varibright):

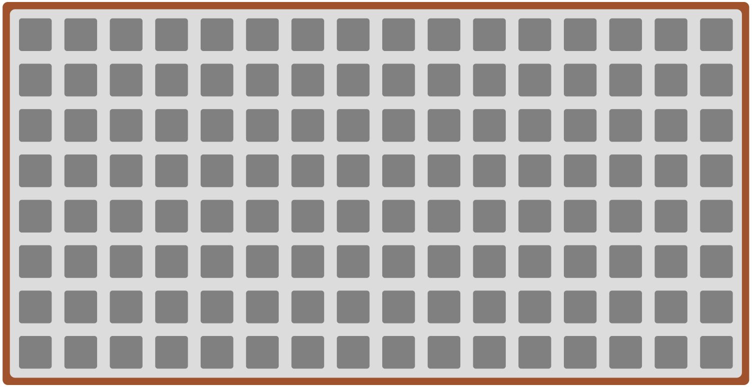
- A matrix of 128 programmable button;
- 16 levels of brightness per button;
- Serial/RS232 (FTDI) connection.

Controlling a "smart" bulb (LIFX Original 1000):

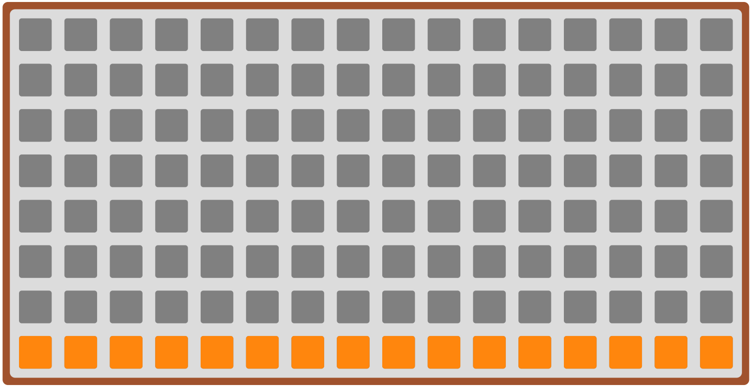
- A ~1000 lumens programmable light bulb;
- Nice colors, nice range of whites;
- Wi-Fi 802.11bgn, 2.4gHz.

Let's have a look at the controller UI.

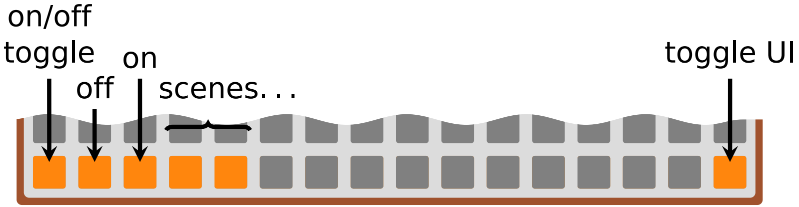
The grid



General functions/scenes row



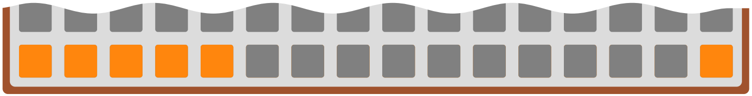
General functions/scenes row



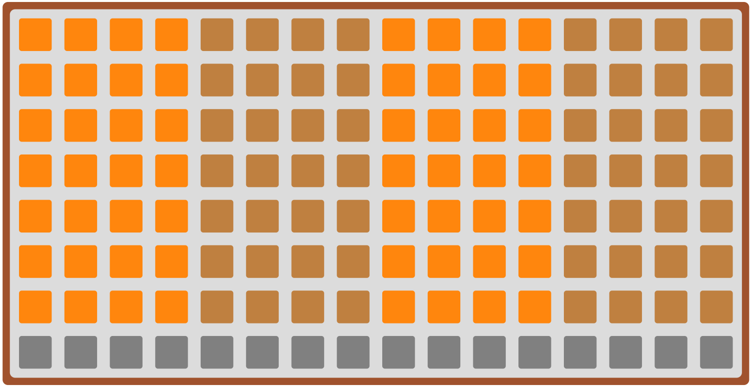
General functions/scenes row

Other ideas:

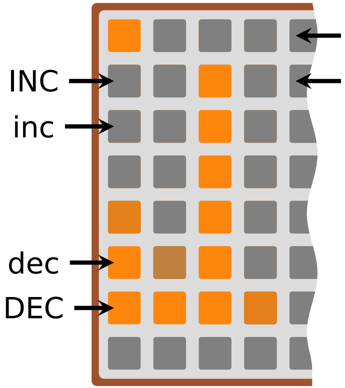
- Navigation controls (pagination. . .);
- MPD control.



Target control pads x4



Target control pads x4



Functions/status row (toggle, TBD. . .)

4 sliders (Hsbk, “color wheel”):

- Hue: 0.0–360.0°;
- Saturation: 0.0–1.0;
- Brightness: 0.0–1.0;
- Temperature: 2500–9000K.

monolight live

- Controls;
- UI feedback;
- monolight layer definitions.

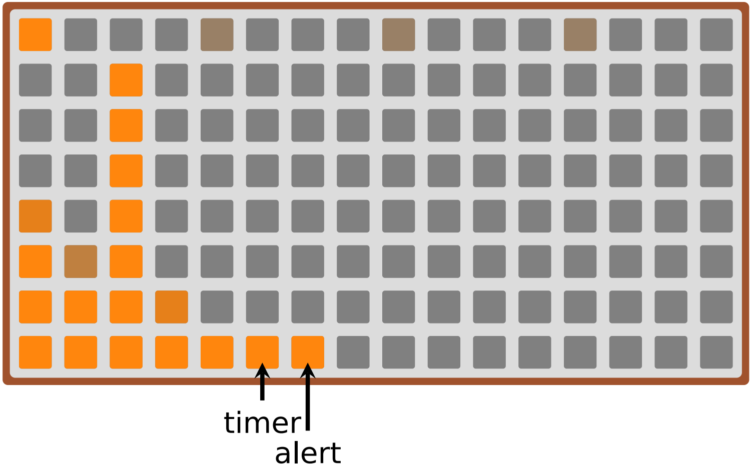
monolight live

- Controls;
- UI feedback;
- monolight layer definitions.

One last (unimplemented) idea I wanna show you. . .

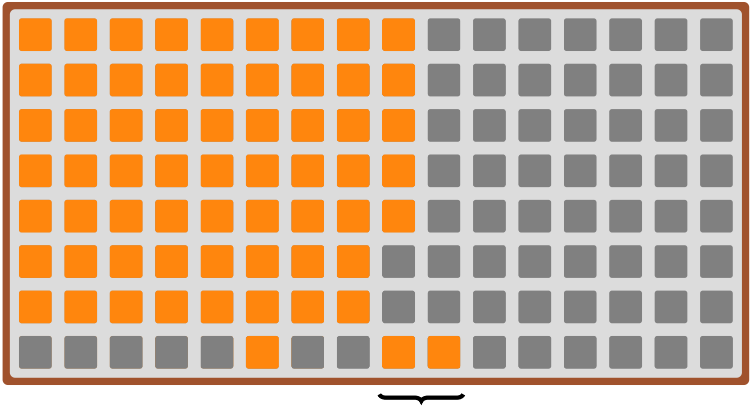
Timer/Alert effect

Let’s add two more functions:



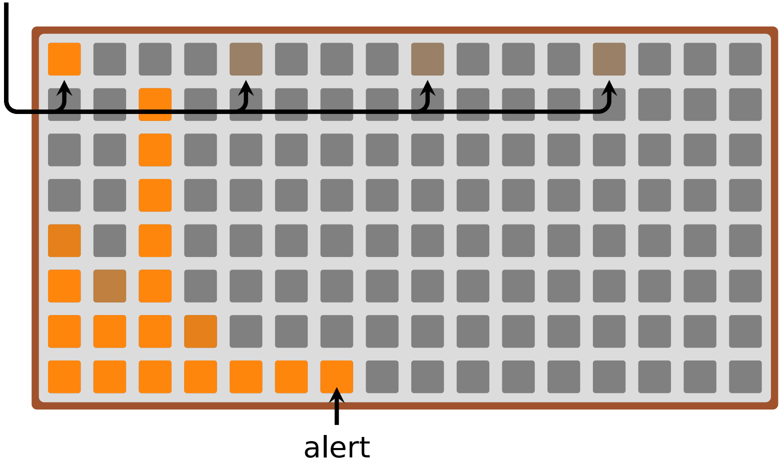
Timer/Alert effect

Time selection (1 lit button = 1 unit of time):

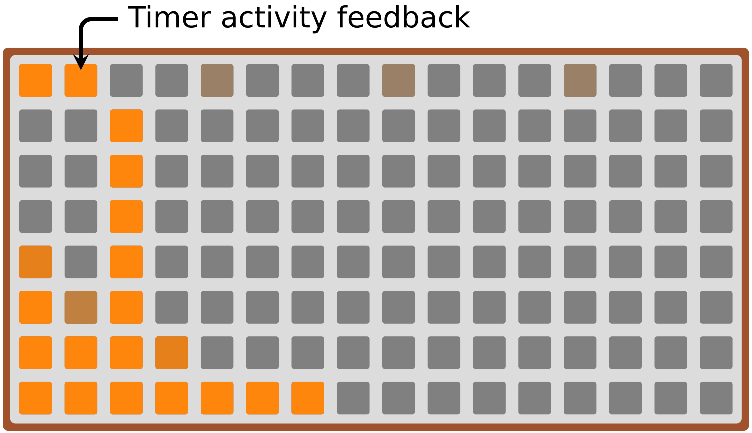


Timer/Alert effect

Target and alert selection:



Timer/Alert effect



monolight implementation

High-level details:

- Python $\geq$ 3.5 (pondering $\geq$ 3.6);
- Fully async (using *asyncio* with the stream API);
- Fully typed, it’s great;
- Very slow, no tests 😐;
- Uses Artem Popov’s *pymonome/aiosc* libraries;
- 2/3 months of work, GPLv3.

monolight implementation

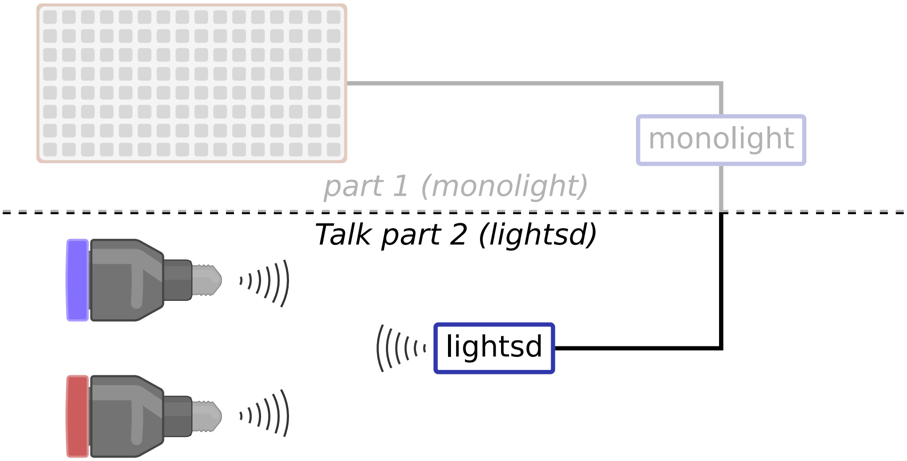
High-level details:

- Python $\geq$ 3.5 (pondering $\geq$ 3.6);
- Fully async (using *asyncio* with the stream API);
- Fully typed, it's great;
- Very slow, no tests 😐;
- Uses Artem Popov's *pymonome/aiosc* libraries;
- 2/3 months of work, GPLv3.

As we've seen, lot of fun stuff left:

- More UI features;
- UI animations;
- Control other things.

High-level architecture



lightsd API live

- get_light_state;
- power_toggle, targeting;
- set_light_from_hsbk;
- set_waveform.

lightsd

The “parent” project:

- C99, libevent2, CMake — that’s all;
- Daemon, low memory footprint, fast enough[1];
- 32/64 bits, big/little endian, FPU optional;
- Runs on nearly everything but Windows[2];
- First PoC in 2014, mostly written through 2015.

[1] A bit of a CPU consumer.

[2] LXSS will fix that though?

lightsd

Original ideas:

- Remove discovery delays and glitches;
- While exposing a high-level *vendor agnostic* API;
- While offering network isolation;
- No cloud nor Internet required;
- GPLv3 with non-GPL users in mind;
- “Accessible”: pretty good C, unit-tests, good docs.

lightsd

Implementation details:

- Uses LIFX's faster and *harder* LAN API;
- Proxies all communications to the bulb;
- Keeps track of the *current* state of the bulbs (sampling);
- High-level API in JSON-RPC over TCP, Unix sockets or a named "command" pipe[1].

[1] The pipe is unidirectional: only usable to send commands.

The (🤕 | 😄) parts

In no particular order:

🤕	😄
C	C
asyncio tasks cleanup	Python 3.5+
Buildbot	Continuous integration
Portability	“Stack position”
Wi-Fi	Playing with the lights
Reverse engineering. . .	in reasonable amounts
Firmwares bugs	User feedback
OS Packaging	

“Stack position”

One thing I really like:

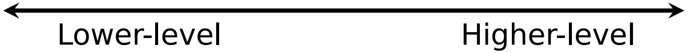
LIFX	**lightsd**	**monolight**
hardware	daemon	GUI
embedded	C	Python

Lower-level Higher-level

“Stack position”

One thing I really like:

LIFX	**lightsd**	**monolight**
hardware	daemon	GUI
embedded	C	Python



lightsd opens-up to a wide range of topics.

Notes on the LIFX bulbs

- Get them on sale;
- Best brightness/colors (AFAIK);
- Standby power consumption;
- Cool LAN API, hope they keep it;
- Only Gen 1 (EOLed in 2015) doesn’t crash for me;

Notes on the LIFX bulbs

- Get them on sale;
- Best brightness/colors (AFAIK);
- Standby power consumption;
- Cool LAN API, hope they keep it;
- Only Gen 1 (EOLed in 2015) doesn’t crash for me;
- *Binary blobs suck.*

“My Roadmap”

Things I wanna do:

- Time based releases;
- Better CI/automation;
- “State-enforcement”;
- Effects API and effects plugins.

Not on my roadmap

Things I wanna have:

- JSON-RPC extensions: streaming, auth, server notifs;
- A reversed-engineered LIFX firmware;
- A firmware that doesn’t crash;
- Support for other brands (Hue?);
- Color calibration;
- LIFX stripe support.

Thanks

Time for Q&A and discussion

- @1opter
- *#lightsd* on IRC (*chat.freenode.net*)
- `https://www.lightsd.io/`

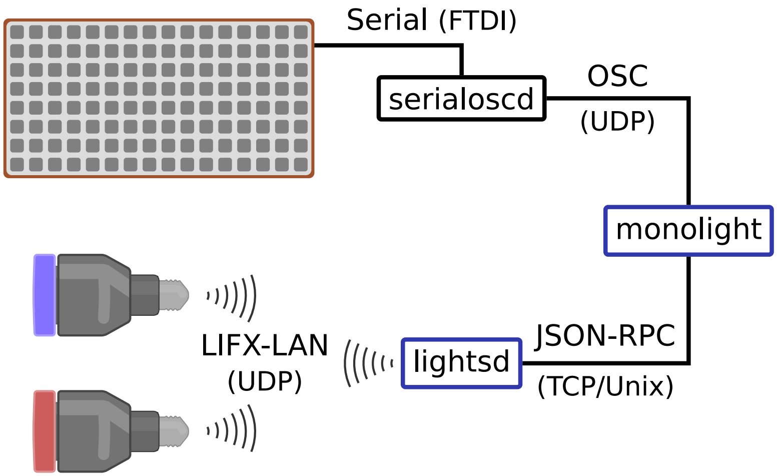
Links

LIFX	website, forum, github;
lightsd	docs, sources, downloads;
monolight	sources;
monome	website, forum, github;
pymonome	sources.

Questions for you!

- Hardware hacks?
- UX with other projects and products?
- “Education” opportunities opinions?

Detailed architecture



LIFX products tables

Generation	**Models**	**Available**
Gen 1	Original 1000, Color 650	No
Gen 2	Color 1000, White 800	Yes
Gen 3	A19, BR30, Z (stripe)	Yes

Generation	**Notes**
Gen 1	Has 802.11 and (unused) 802.15.4
Gen 2	QCA 4002, AllJoyn, *crashes*
Gen 3	+ versions have IR, *still crashes*