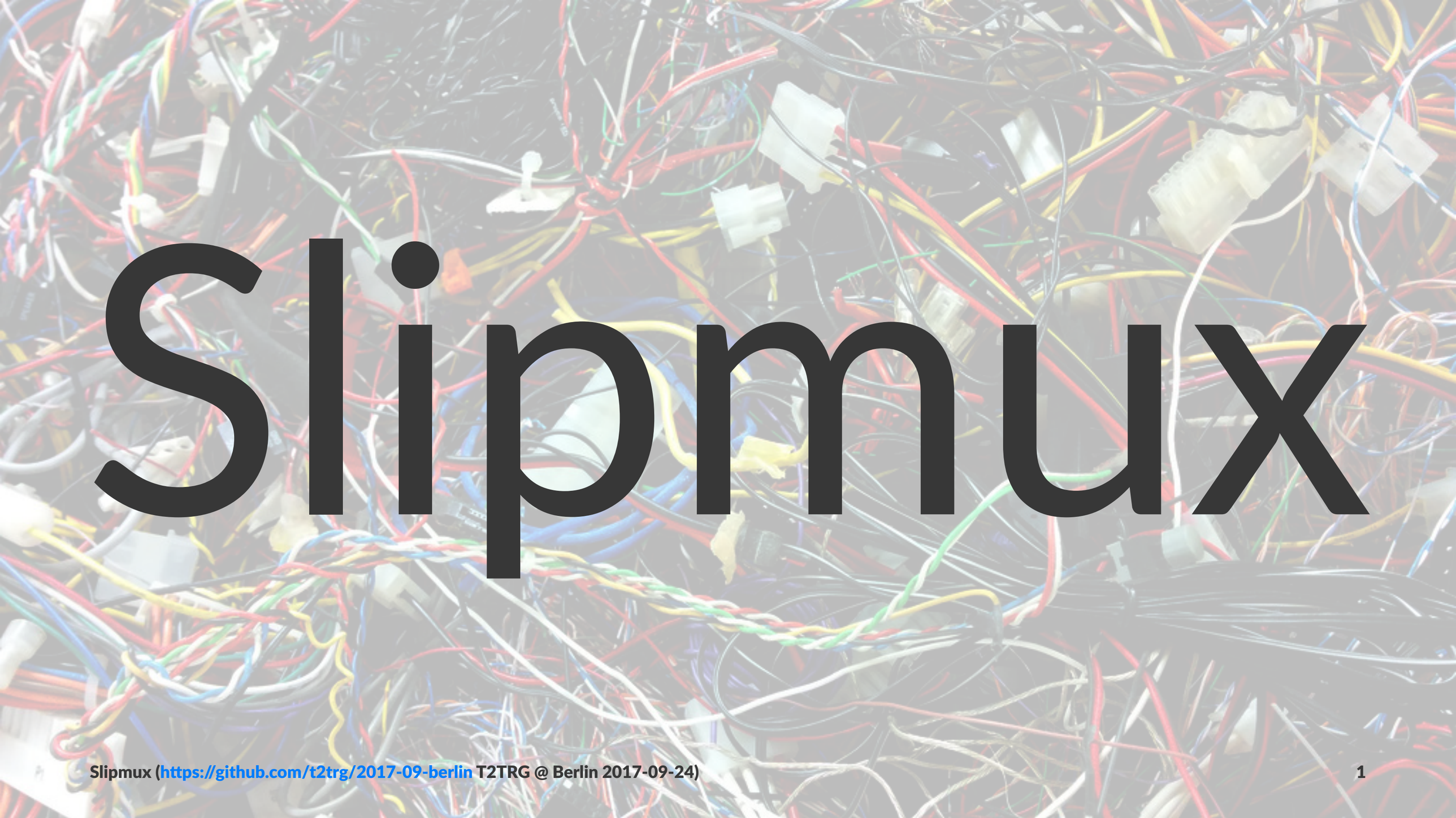
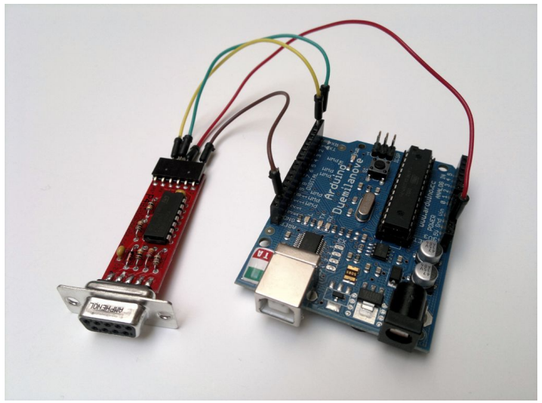

Slipmux

One UART to bind them all



Use the laptop to:

- Control the microcontroller
  - Flashing
  - Setup, Configuration
  - Operational control
- Get diagnostic output

Use a serial interface for packets

There have been packet interfaces based on SLIP

- e.g., Contiki tunslip
- RIOT ethos

Usually with an ad-hoc, non-extensible control hack

➔

Need separate serial for other control functions

Slipmux:

What's New?

Slipmux approach:

More aggressive multiplexing

Combine:

- Packets over serial (IPv4, IPv6)
  - add Ethernet, etc., as needed
- A diagnostic channel ("tty" style, out only)
- CoAP for control

Diagnostic channel

Kernel printfs

Used as a low-threshold way to get debug output to a laptop

Output only

- There is no input parser

CoAP for control

Get rid of "shell" or other serial parsers

Define CoAP endpoints to replace shell commands

```
POST coap+uart:///ifconfig/5/add/2001:db8::3
```

Can now easily run control from a Makefile

Can extend this to return structured data

Technical details

Typical:

- "TTL" or 3.3 V levels
- 115200 bit/s, 8N1

Add a 16-bit CRC for CoAP exchanges

- CoAP then handles the reliability

Implementation: RIOT side

Replace *shell_run* by CoAP parser for SLIP input

- For now, use diagnostic channel for printf
- Extend shell_handler interface by return path

Funnel printf output back into SLIP frames

- Needs some synchronization, of course
- MTU HOL blocking at 115200 bit/s: 130 ms
  - Could abort (and restart?) IP frames instead

Implementation: laptop side

Needs a *slipmuxd*

- Replaces pyterm etc.
- Handles IP packets
- Provides diagnostic output
- Provides a minimal CoAP proxy
  - Could even provide a shell-like input parser

Written up:

`draft-bormann-t2trg-slipmux`

(Carsten Bormann, Tobias Kaupat)

To do:

- Define stable CoAP endpoints
    - Many of which will be OS specific

How far do we take this

Flashing over CoAP?

- Could cross-breed with OTA functions

Gdb access?

- Well, maybe not
- It is still nice to only have to set up SWD/JTAG for serious debugging

Do implementers like the idea?

Is this something that could be *standardized*?

Is this something that *should* be standardized?

Acknowledgments

Thank you, Tobias Kaupat

Images

Cory Doctorow [cabletangle], Makezine: [arduinoserial]

[cabletangle] https://www.flickr.com/photos/doctorow/14452760238

[arduinoserial] http://makezine.com/wp-content/uploads/make-images/b6Jj2cwPWvgIYGWn.jpg