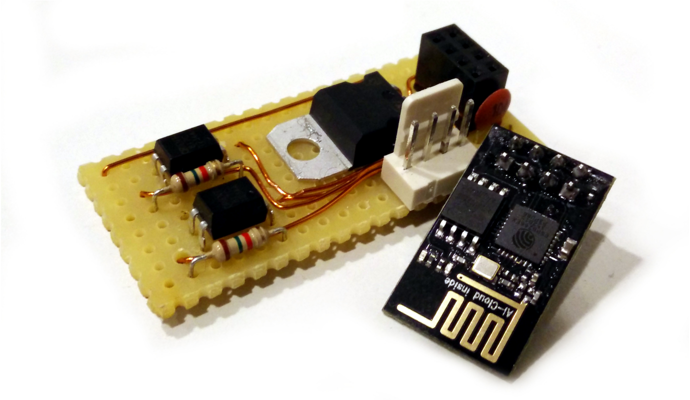
Implement Updates *Over the Air* for *ESP8266* microcontrollers

Who? Dustin Frisch
C/C++ for Embedded Systems and Physical Computing
Angewandte Informatik
University of Applied Sciences Fulda
`dustin.frisch@informatik.hs-fulda.de`

When? February 9, 2017

About the Project

ESPer



About the Project

Magrathea Laboratories e.V.

Updates

- Changed requirement or environment
- Closing security issues
- Fixing bugs
- New Features

Often possible without hardware changes.

Over the Air (OTA)

Reduces maintenance cost.

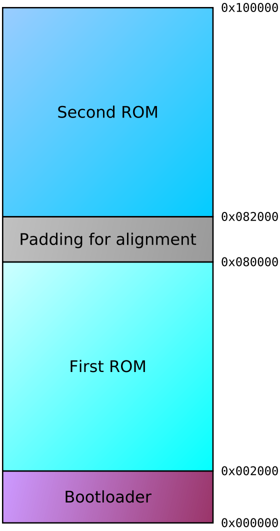
Requirements

- Automatic updates *OTA* without administrative interaction
- Insusceptible to errors of any kind
- Downloading side-by-side with payload traffic
- Maximal uptime and minimal network load
- Maintenance and monitoring
- Categorizing devices by task-specific hardware

Implementation

Update mechanism

1. Download `.version` file and compare
2. Download firmware binary for *other* slot
3. Reconfigure *rBoot* bootloader
4. Restart



Implementation

Supporting multiple devices

```
#include "Device.h"
#include "features/Socket.h"

Device device;

constexpr const char SOCKET_NAME[] = "socket";
constexpr const uint16_t SOCKET_GPIO = 12;
OnOffFeature<SOCKET_NAME, SOCKET_GPIO, false, 1>
    socket(&device);

Device* getDevice() { return &device; }
```

Implementation

Supporting multiple devices

```
make socket/flash
```

to build and flash.

Implementation

Automatic deployment and roll-out

- `git commit`
- *drone* builds it
- `scp *.version *.rom{0,1}` to deployment server
- Served by *nginx*

Conclusion

It works!

`https://git.maglab.space/esper/esper`

Thank you for your attention.