

Zephyr & Thingsboard

Zephyr Project Meetup

05.06.2025

Christian Hirsch

christian@hirsch.zone

About me

- **Self-employed**
 - Hardware- and Software Development
 - Developing *digidrip.eu*
- **Passionate about**
 - Embedded Design
 - Open Source
 - Brewing Beer
 - Farming



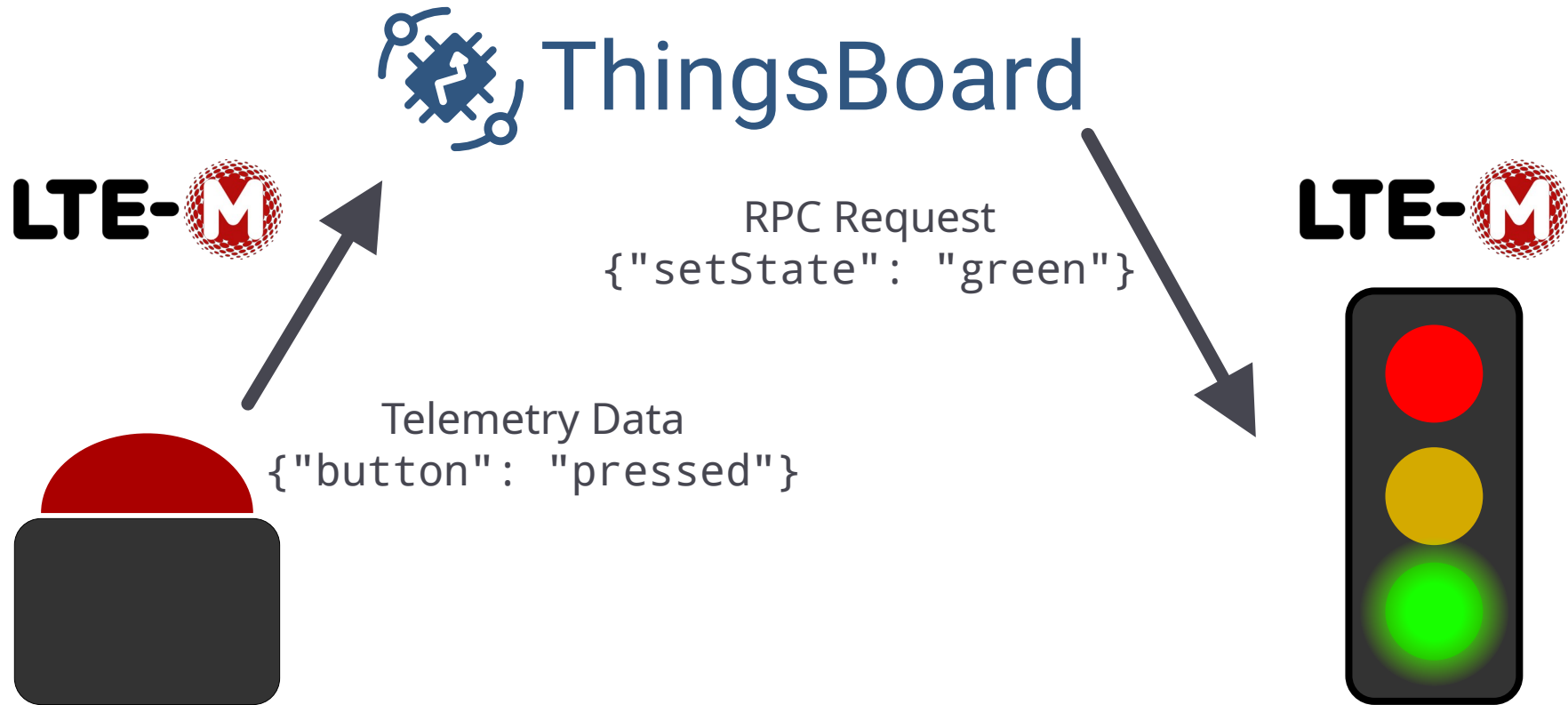
Agenda

- Use-case / Example
- Thingsboard
- Zephyr Implementation
- Live Demo



Traffic Light Control

Architecture Overview

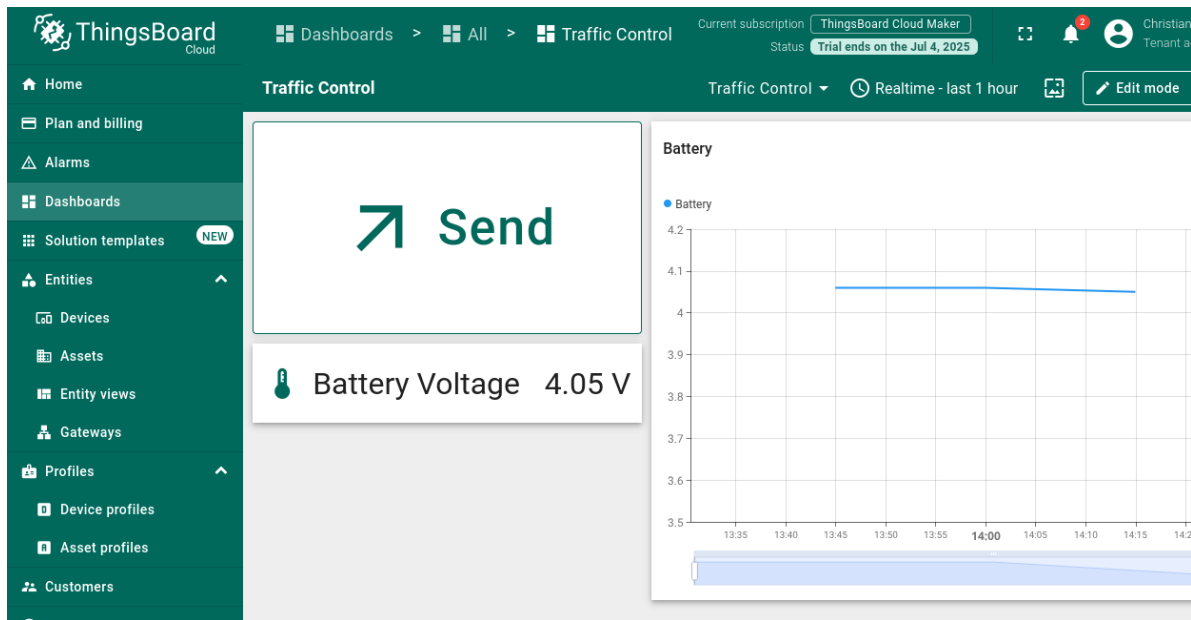


Thingsboard

- Open-Source IoT Platform
- Device Management
- Data Collection and Processing
- Visualization

Thingsboard

- Telemetry Data
 - Measurements
- Attributes
 - Client
 - Server
 - Shared
- RPCs



Thingsboard API

- MQTT
- CoAP
- HTTP
- LWM2M
- SNMP

Telemetry via MQTT

- Publish to "v1/devices/me/telemetry"
- {"button": "pressed"}

zephyr_009CF266

Device details

?

×

<

Details

Attributes

Latest telemetry

Calculated fields

Alarms

>

Telemetry

+

🔍

☐	Last update time	Key ↑	Value	
☐	2025-06-04 16:02:35	button	pressed	



RPC via MQTT

- Subscribe to
`"v1/devices/me/rpc/request/+"`
- ```
{
 "method": "setState"
 "params": {"targetState": "green"}
}
```

# RPC via MQTT with TB Widget

On click



Action

Execute RPC



Method\*

setState

Parameters

Constant

Function

None

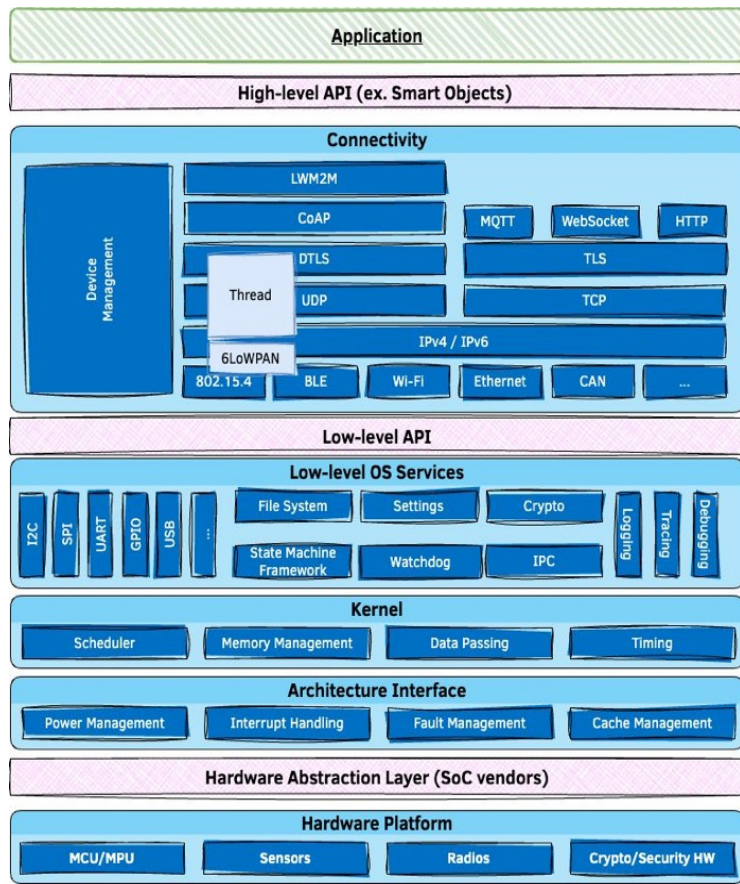
{...} JSON



{"targetState":"green","timeout":5}

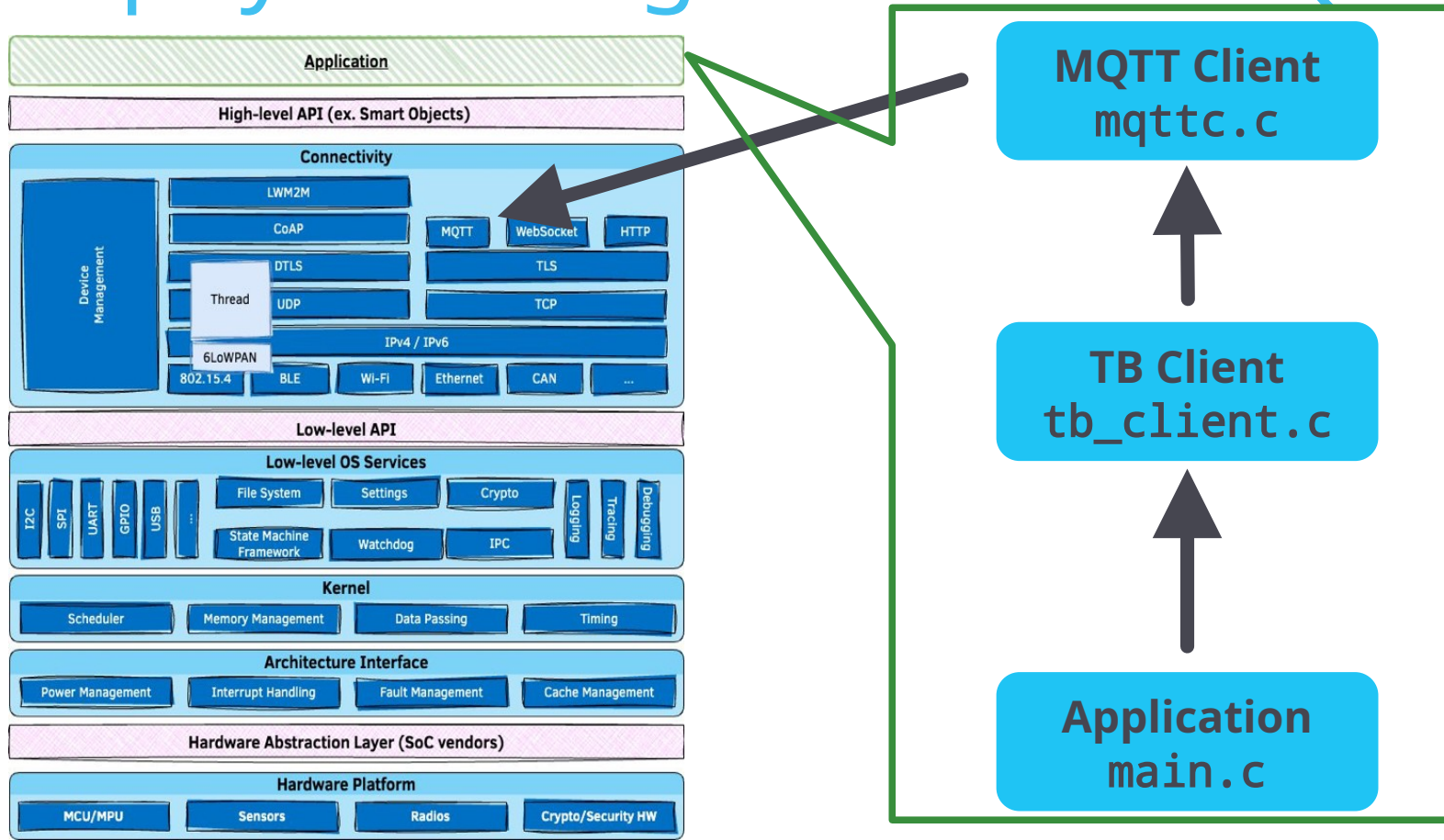


# Zephyr + Thingsboard



- Connect via
  - MQTT
  - CoAP
  - LwM2M
  - HTTP
- Using
  - WiFi / Ethernet / LTE-M

# Zephyr + Thingsboard w/ MQTT



# MQTT Client

- Connect to Broker
- Handle Connection
- Publish Messages
- Subscribe to Topics

# Zephyr Thingsboard Client

- Uses MQTT Client
- Provisions Devices
- Sends
  - Attributes
  - Telemetry
- Receives
  - Attributes
  - RPC Requests

# Telemetry Upload

```
/* tb_client.h */
```

```
int tb_client_telemetry_publish(uint8_t *_data);
```

```
/* main.c */
```

```
void button_pressed_work_handler(struct k_work *work)
{
 ...
 tb_client_telemetry_publish("{ \"button\": \"pressed\" }");
 ...
}
```



# RPC Handling

```
/* tb_client.h */
```

```
int tb_client_rpc_subscribe(const char *method, int(*callback)(uint8_t *,
size_t));
```

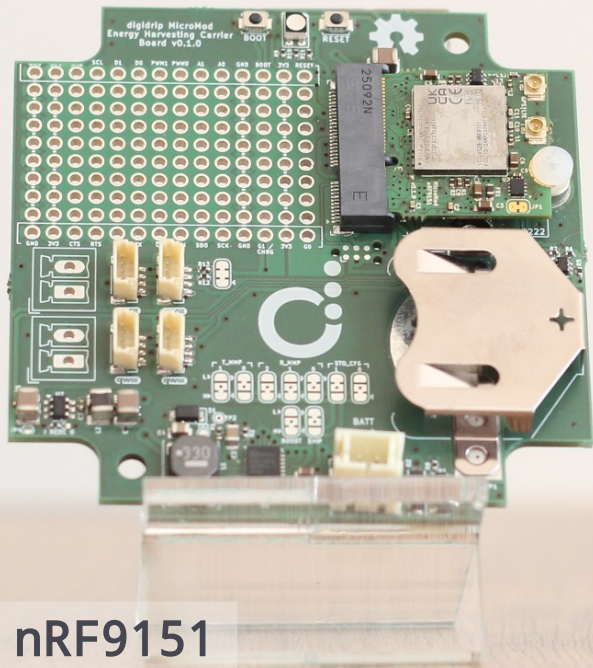
```
/* main.c */
```

```
void rpc_set_state_cb(uint8_t *message, size_t len) {
 /* handle RPC request */
}
```

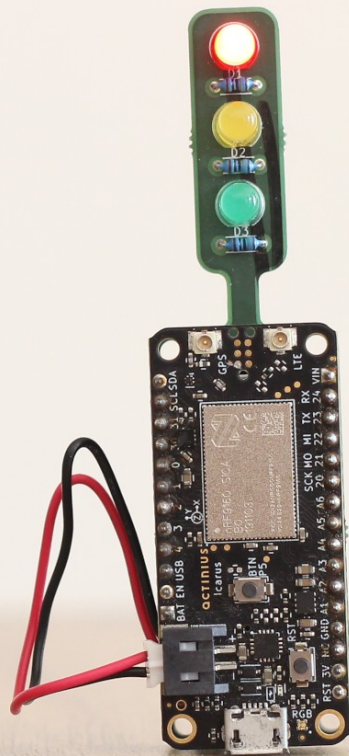
```
...
```

```
int main(void) {
 tb_client_rpc_subscribe("setState", &rpc_set_state_cb);
}
```

# Example / Hardware



MicroMod nRF9151  
Processor Board



Actinius Icarus  
• nRF9160

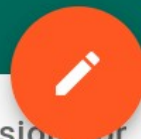
A close-up photograph of a black traffic light housing. Inside the circular lens, a green bicycle symbol is illuminated. The word "Showtime" is written in a white, bold, sans-serif font, centered over the bicycle symbol. The background is a blurred street scene with a crosswalk and some orange traffic cones in the distance.

**Showtime**

# Thingsboard Relations

zephyr\_009CF266

Device details



< Related fields

Alarms

Events

**Relations**

Audit logs

Version control >

Outbound relations

Direction

From



Type ↑

To entity type

To entity name



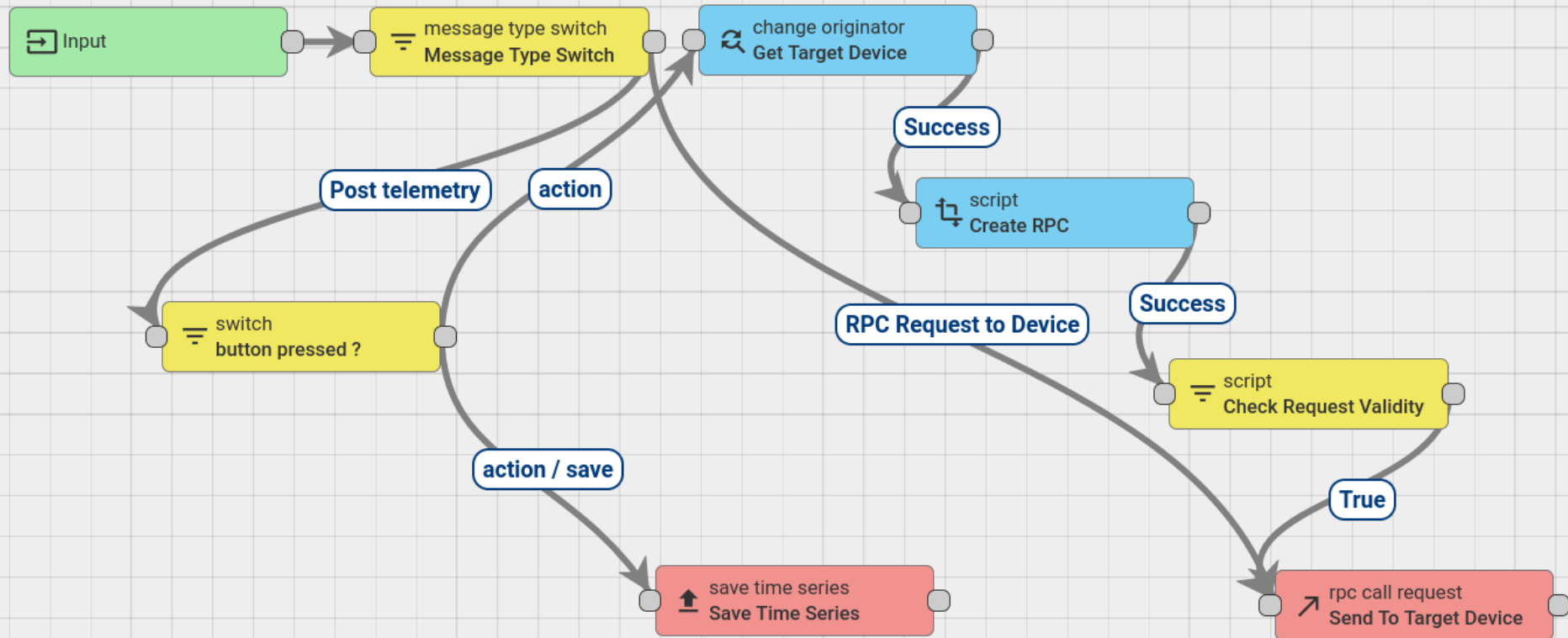
Uses

Device

[zephyr\\_307D069A](#)



# Thingsboard Rulechain





A close-up photograph of a black traffic light. The central lens is illuminated with a green light, showing a green bicycle symbol. Overlaid on the center of the bicycle symbol is the text "Showtime II" in a white, bold, sans-serif font. The background is a blurred street scene with a crosswalk and some orange traffic cones in the distance.

# Showtime II

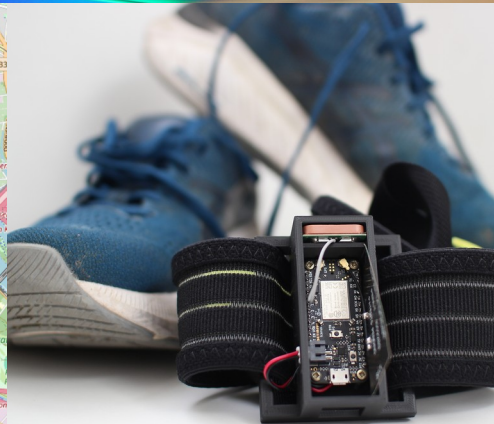
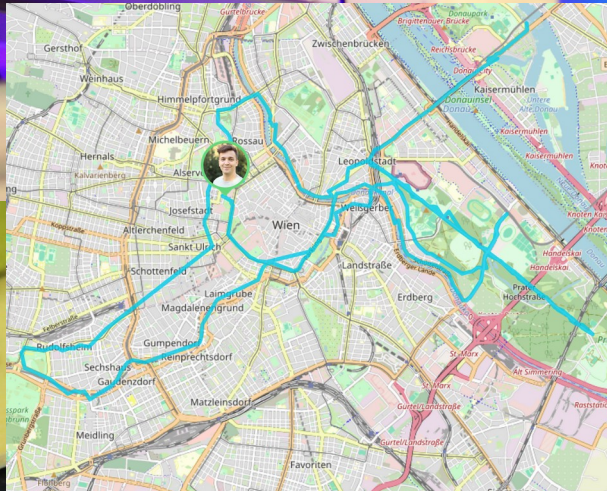
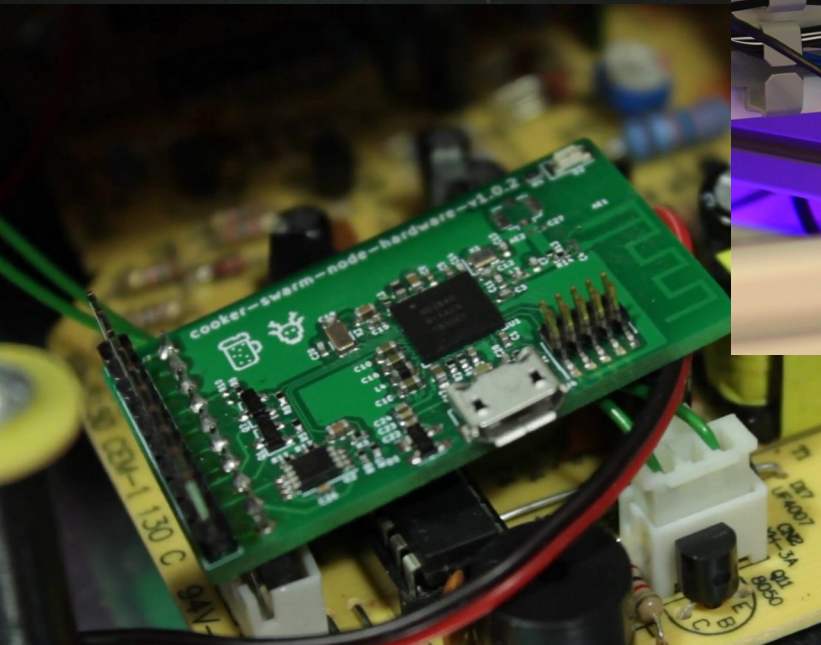
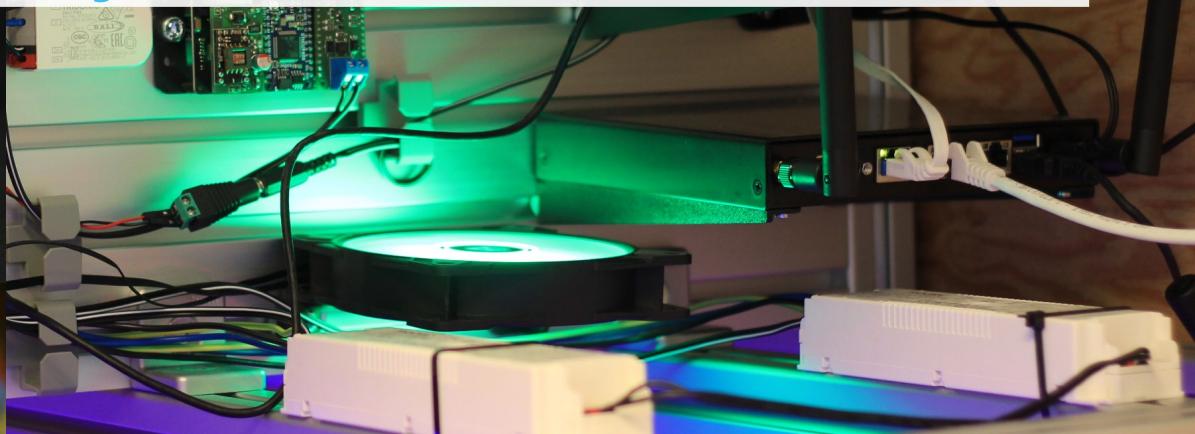
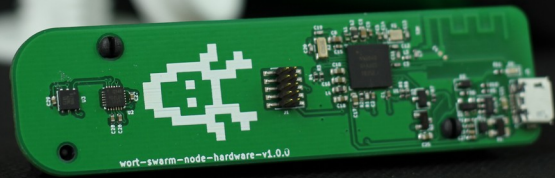


# Real-World Example: Irrigation





# Examples and Projects



Ready for Running



# Kontakt

## Christian Hirsch

Hardware / Software Entwicklung, IoT

E: christian@hirsch.zone

W: hirsch.zone

- LinkedIn
- GitHub

