# Observing a Moving Target
# Reliable Transmission of Debug Logs from Mobile Embedded Devices

Björn Daase, Leon Matthes, Lukas Pirl, Lukas Wenzel
Operating Systems and Middleware Group, Hasso Plattner Institute
TDIS 2021 | 03.11.2021

# Mobile Embedded Devices

- Over 98% of all manufactured microcontrollers are embedded
- Moving embedded devices are harder to debug
  - Connectivity is limited
  - Energy budget typically fixed

[1]

[2]

[3]

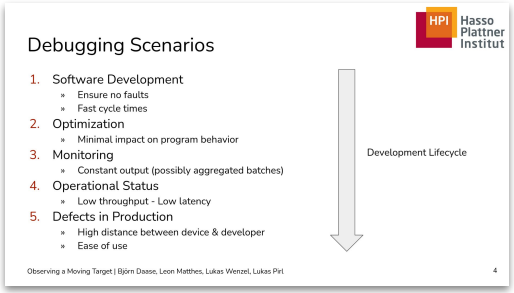
[1] https://commons.wikimedia.org/wiki/File:Volkswagen_ID.3_at_IAA_2019_IMG_0212.jpg
[2] https://pixabay.com/de/photos/iphone-smartphone-apps-apple-inc-410324/
[3] https://www.pngmart.com/files/13/Smartwatch-PNG-Free-Download.png

# The Importance Of Debugging

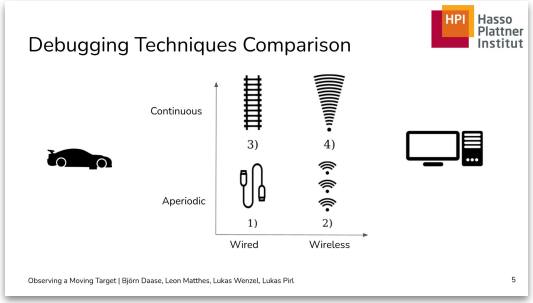
- ~50% of development time spent debugging

[1]

- Debugging is involved in the entire development lifecycle

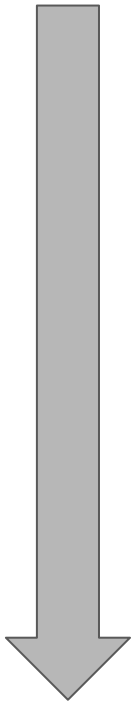



- Communication is main activity in debugging



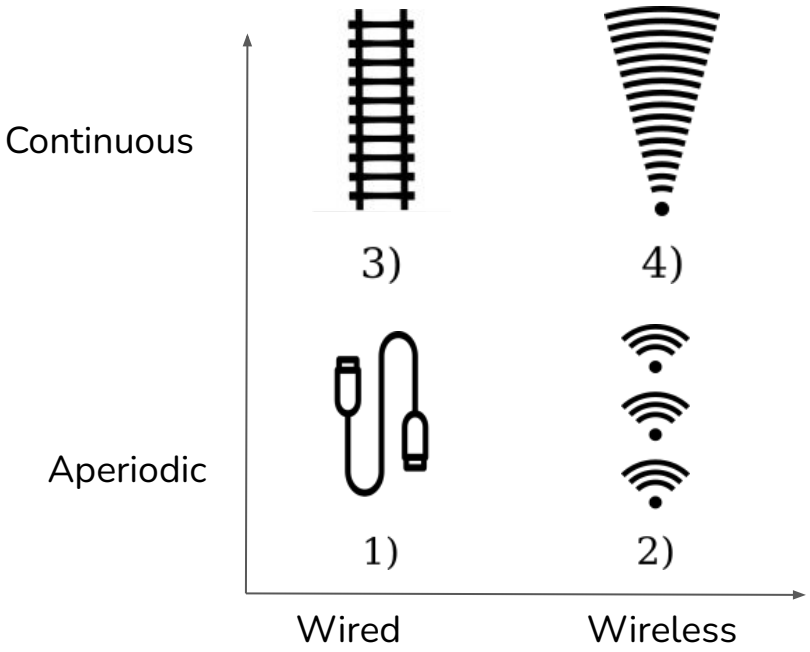
# Debugging Scenarios

1. Software Development
   - » Ensure no faults
   - » Fast cycle times
2. Optimization
   - » Minimal impact on program behavior
3. Monitoring
   - » Constant output (possibly aggregated batches)
4. Operational Status
   - » Low throughput — Low latency
5. Defects in Production
   - » High distance between device & developer
   - » Ease of use



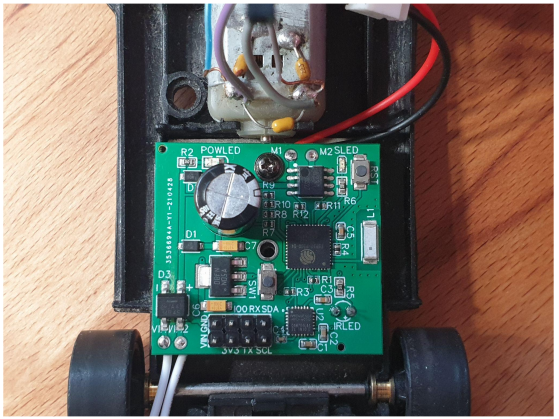
Development Lifecycle

# Debugging Techniques Comparison

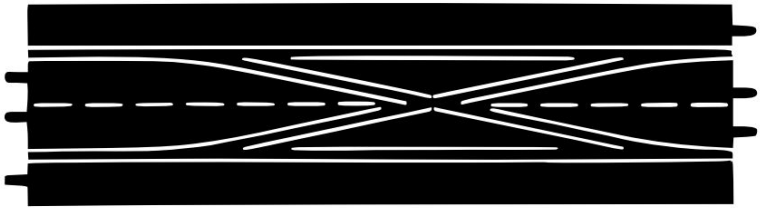


# Case Study - Self-Driving Slot Car

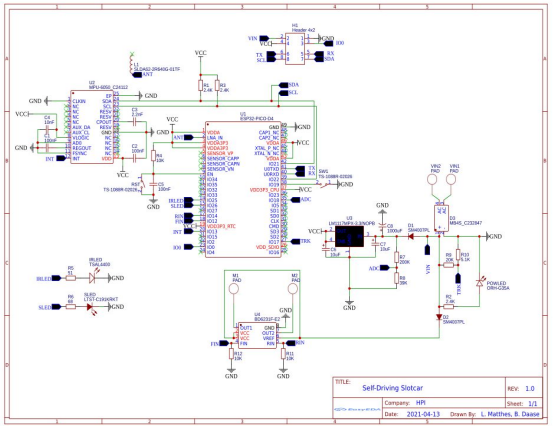
- Programmable slot car based on ESP32 microcontroller
  - Available connectivity: UART/USB, WiFi & BLE
  - Hard real-time constraints to avoid derailing



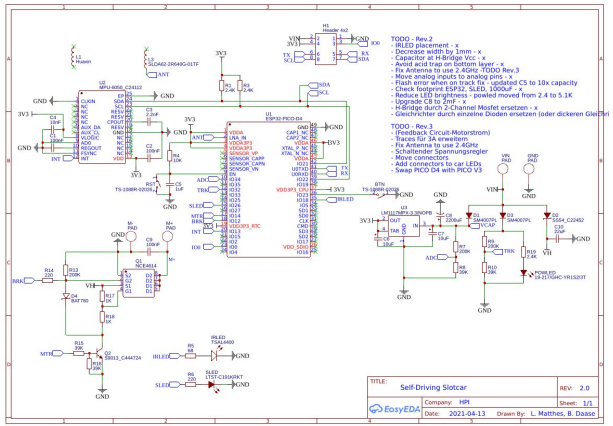

- Energy constraint: Lane change segments
  - Power outage for ~20 ms
  - 1000µF Capacitor



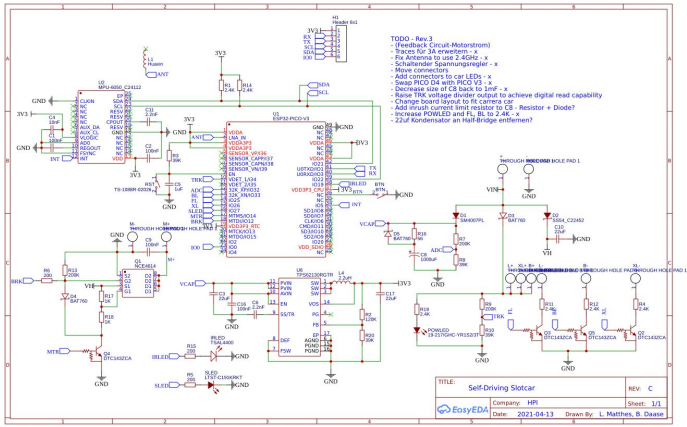
# The Circuit Over Time...



Revision 1 - ☠️

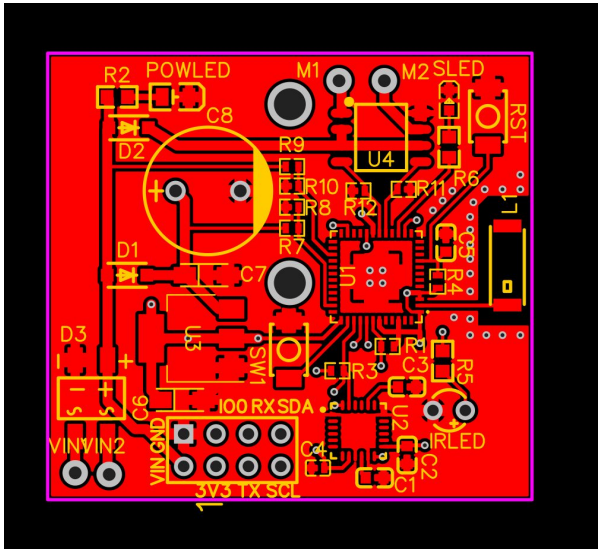


Revision 2 - ✅

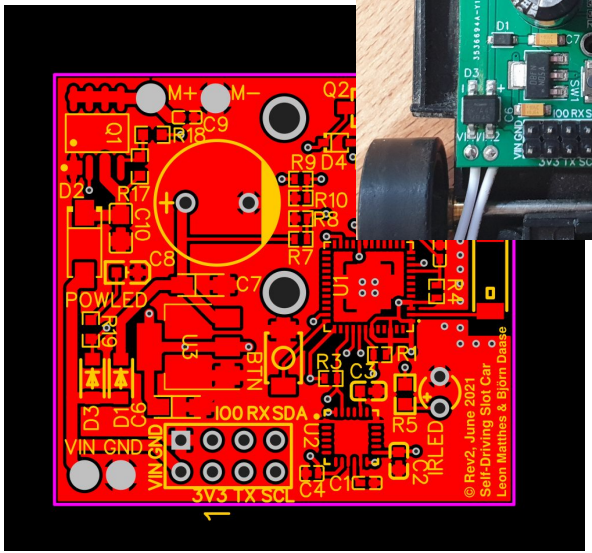


Revision 3 - ⏰

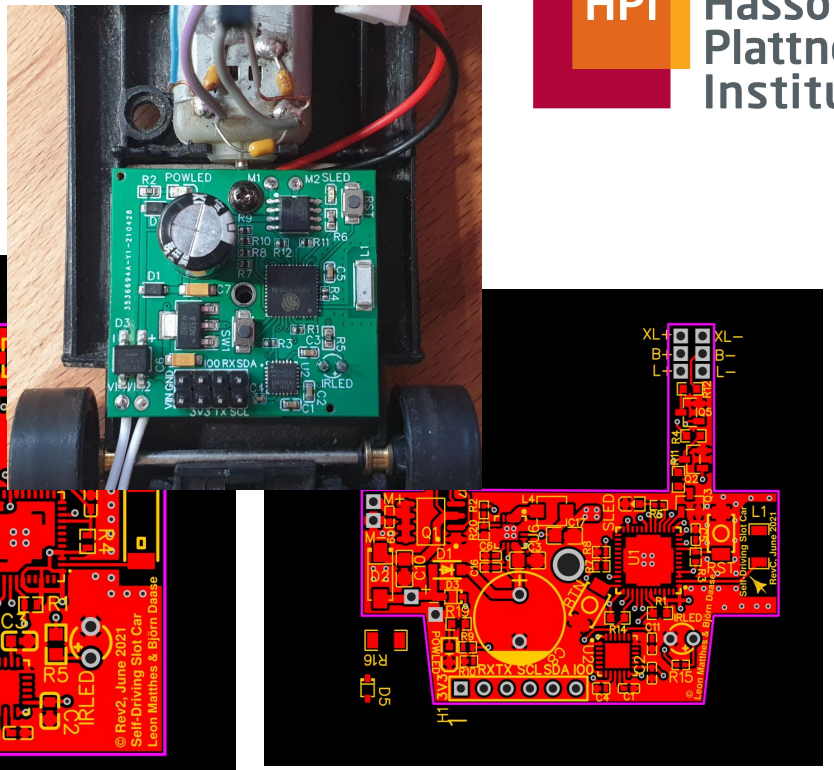
# The PCB Over Time...




Revision 1 - ☠️




Revision 2 - ✅



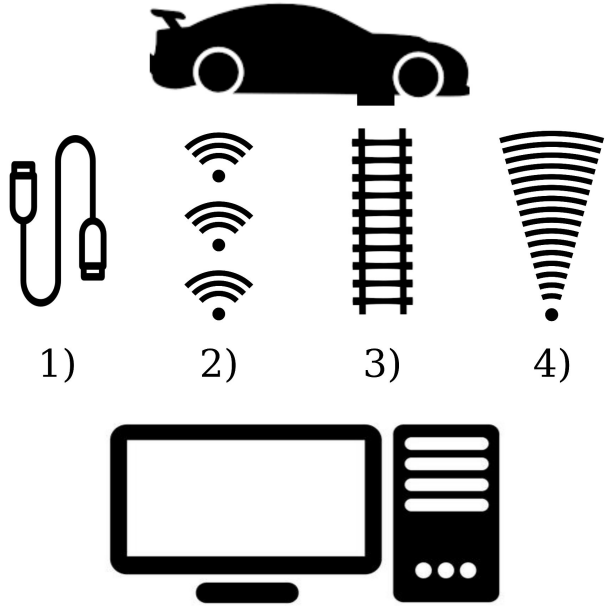

Revision 3 - ⏰

# The PCB Over Time…

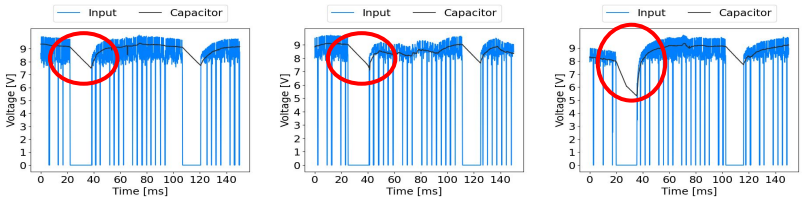
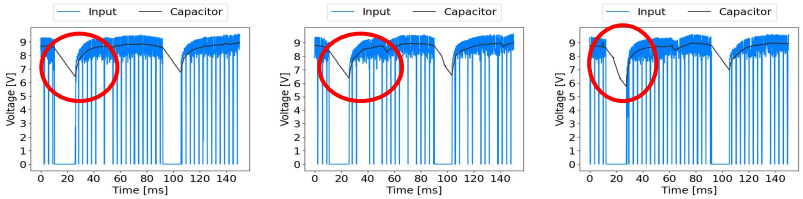
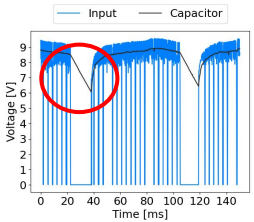
- Revision 1:
  - Initial Design
- Revision 2:
  - H-bridge -> Half Bridge
  - Improved Power Delivery (12 instead of 10 Volt)
  - Minimal downsizing, slight repositioning of components (esp. IRLED)
- Revision 3:
  - Full Redesign (Electrical as well as spacial)
  - Should fit into 4/7 cars in HPI Lab
  - Light Control
  - Further Improved Power Delivery -> Smaller Lane Change Capacitor

# Debugging Technique Application

1) Save & Print Later
   - » Easy to set up
   - » Low performance impact
2) Stop & Radio
   - » Stopping anywhere on the track
3) Write to Carrera track
   - » Continuous output
   - » Extremely low data rate
4) WiFi On-the-fly
   - » Continuous output
   - » High power consumption — Limited performance
   - » Unpredictable



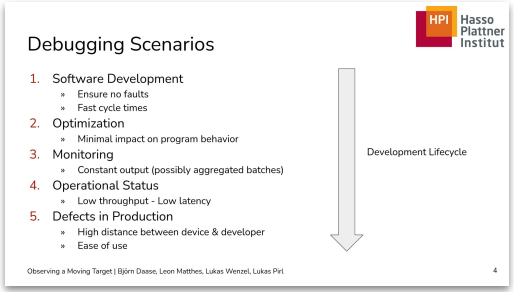
# Energy Consumption

| | No WiFi | Not Sending | Sending |
|---|---|---|---|
| 80 MHz |  |  |  |
| 160 MHz |  |  |  |
| 240 MHz |  | X | X |



- Wireless is more energy intense
- Unpredictable usage spikes
- Problem worsens with increasing computational power
  - Up to complete brownout

# Conclusion

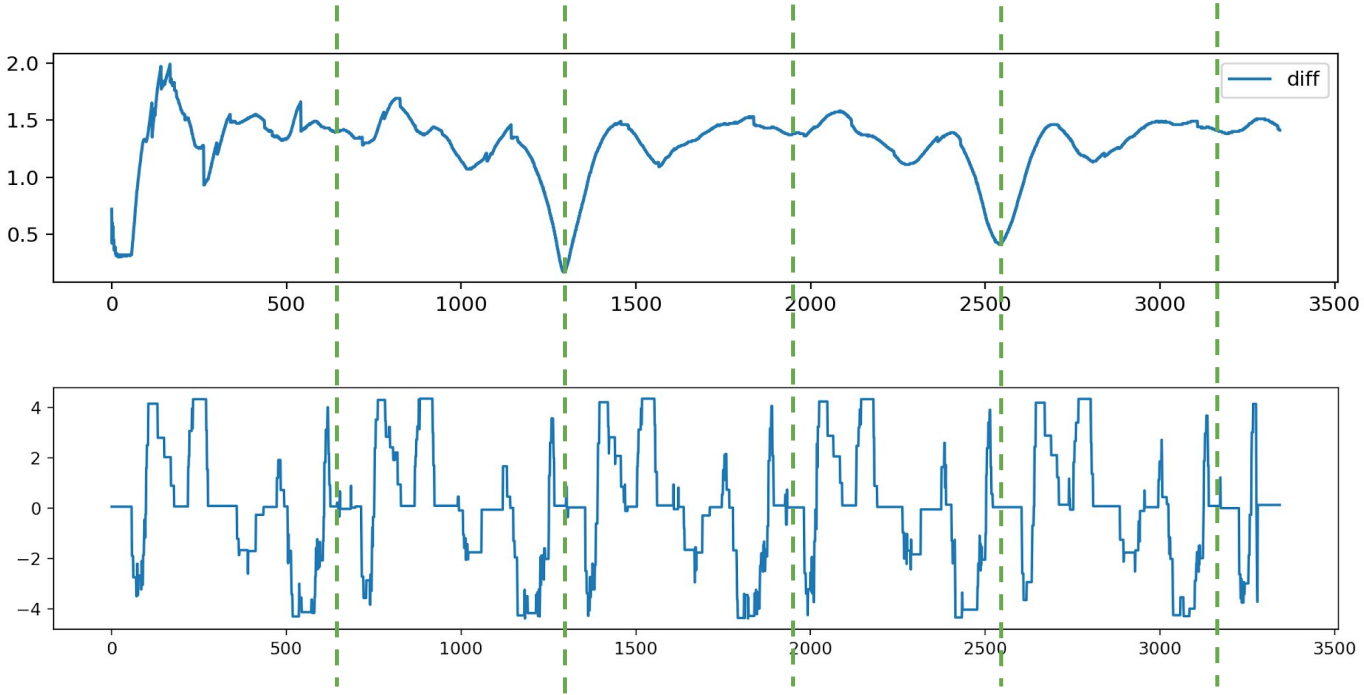


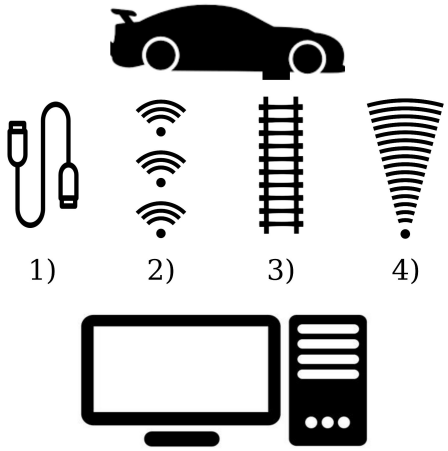

- Wireless debugging
  - \+ offers more applicability & convenience
  - \- Increased efforts for energy budgeting under constrained circumstances

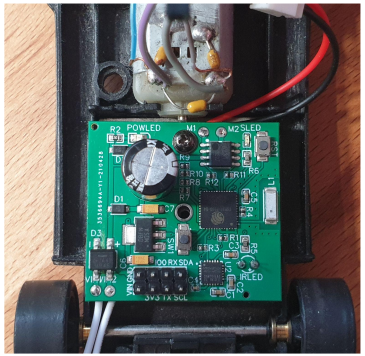
- Available hard- & software modules require more adaptive instrumentation

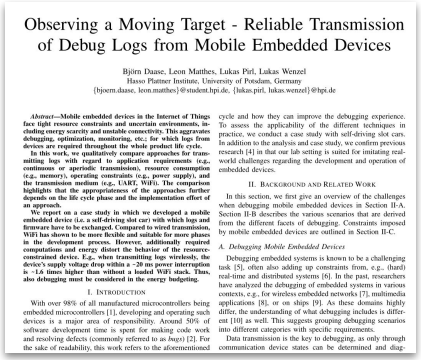
- Automatically select power states based on a configurable energy budget in the future

# Track Detection





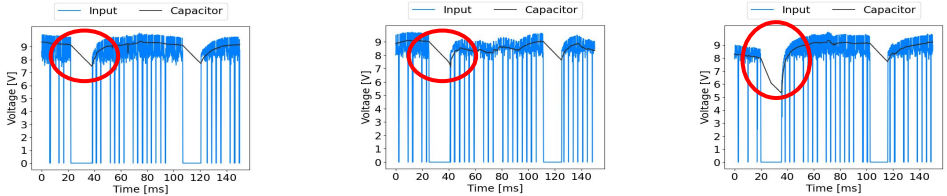





Observing a Moving Target - Reliable Transmission of Debug Logs from Mobile Embedded Devices

Björn Daase, Leon Matthes, Lukas Pirl, Lukas Wenzel
Hasso Plattner Institute, University of Potsdam, Germany
{bjoern.daase, leon.matthes}@student.hpi.de, {lukas.pirl, lukas.wenzel}@hpi.de

# Questions?

https://gitlab.com/hpi-potsdam/osm/self-driving-carrera/

bjoern.daase@student.hpi.de, leon.matthes@student.hpi.de