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

EOS 21/22 | 03.11.2021

HPI Hasso Plattner Institut

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]

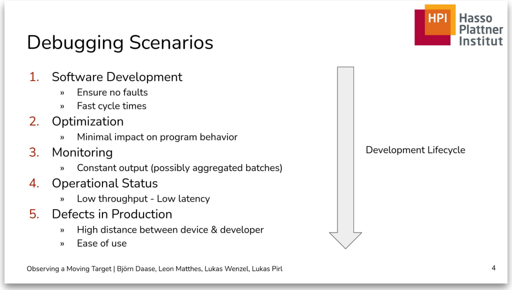
[1] https://commons.wikimedia.org/wiki/File:Volkswagen_ID.3_at_IAA_2019_IMG_0212.jpg
[2] https://pixabay.com/de/photos/iphone-smartphone-apps-apple-inc-410324/
[3] https://www.pngmart.com/files/13/Smartwatch-PNG-Free-Download.png

The Importance Of Debugging

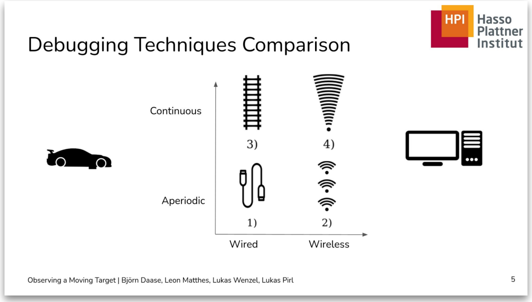
» ~50% of development time spent debugging

[1]

» Debugging is involved in the entire development lifecycle



» Communication is main activity in debugging

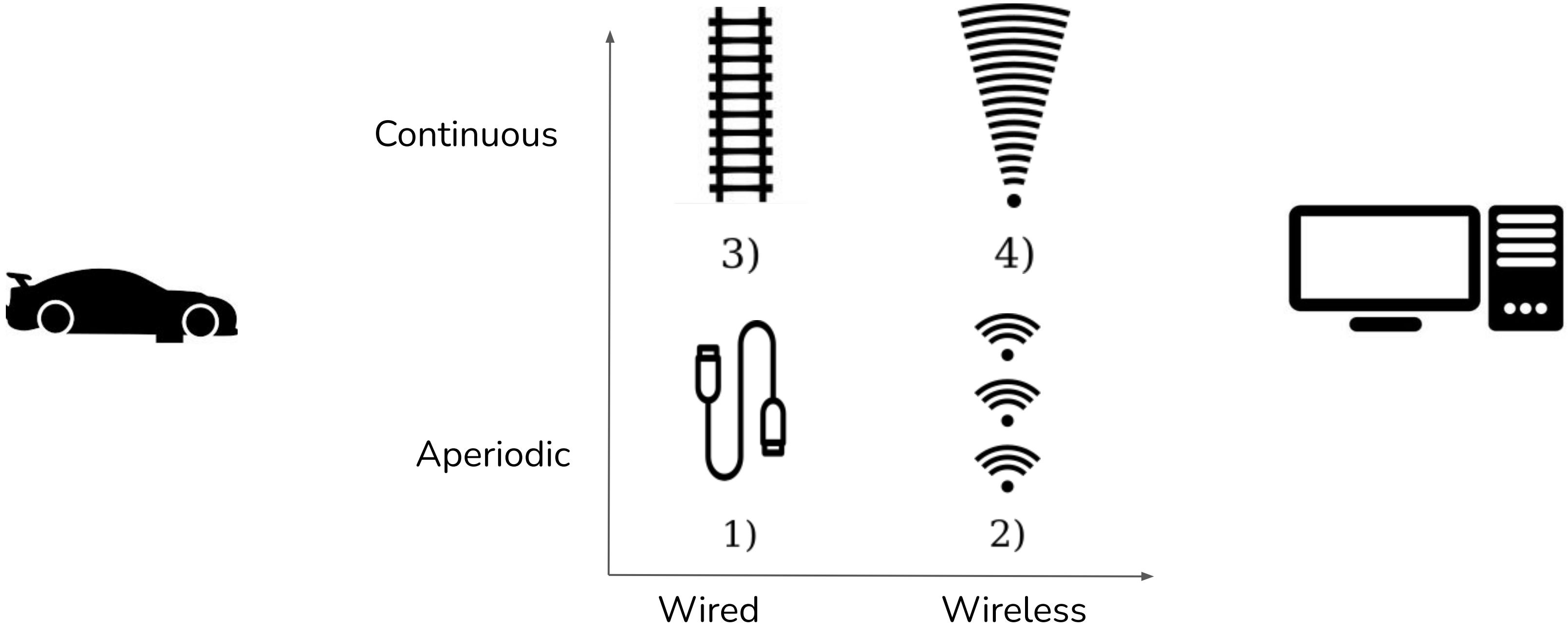


Debugging Scenarios

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

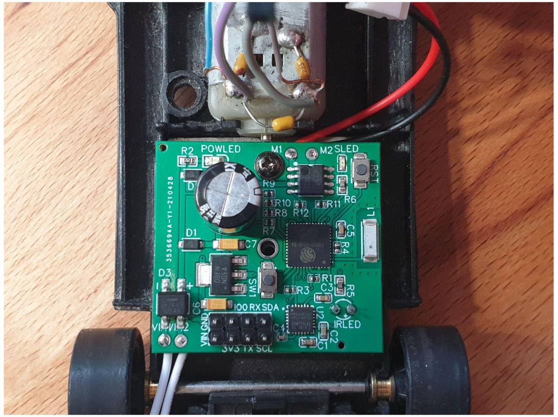
Development Lifecycle

Debugging Techniques Comparison

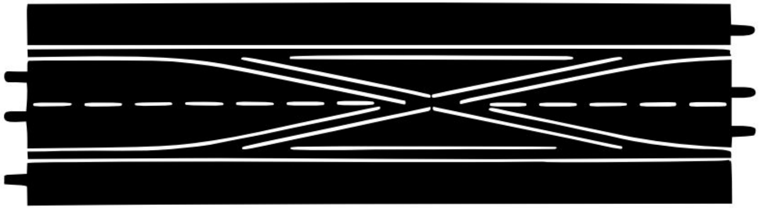


Case Study - Self-Driving Slot Car

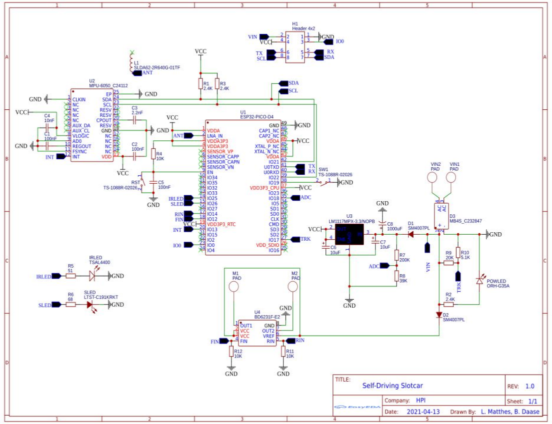
- Programmable slot car based on ESP32 microcontroller
  - Available connectivity: UART/USB, WiFi & BLE
  - Hard real-time constraints to avoid derailing



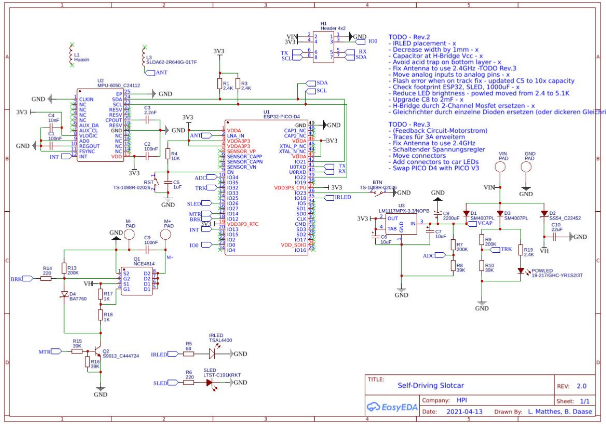
- Energy constraint: Lane change segments
  - Power outage for ~20 ms
  - 1000µF Capacitor



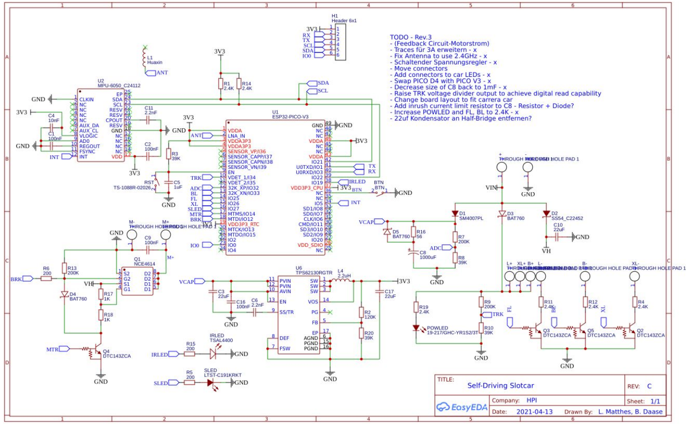
The Circuit Over Time…



Revision 1 - ☠️



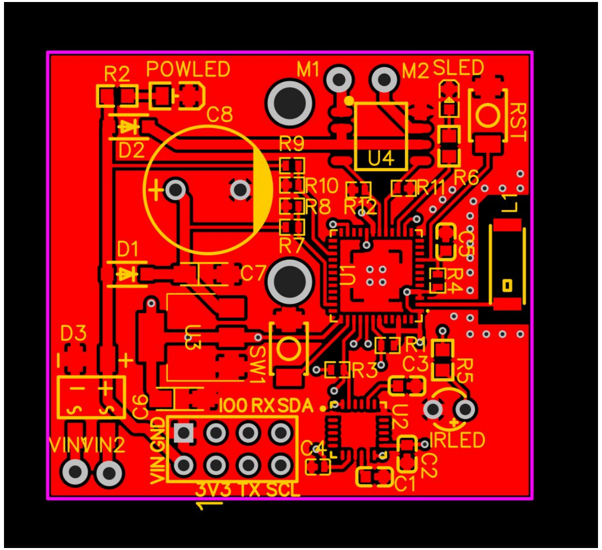
Revision 2 - ✅

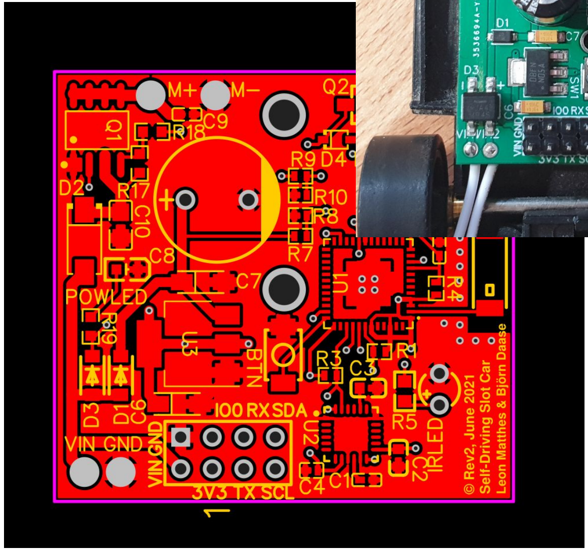


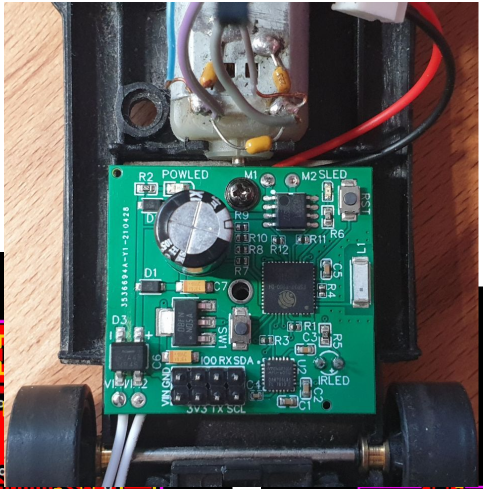
Revision 3 - ⏱️

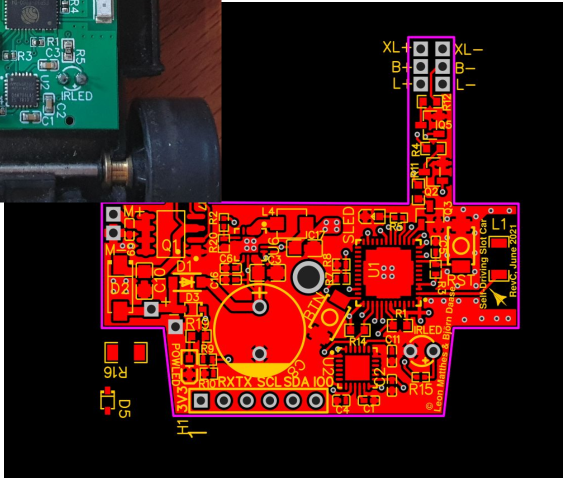
HPI Hasso Plattner Institut

The PCB Over Time…









Revision 1 - ☠️

Revision 2 - ✅

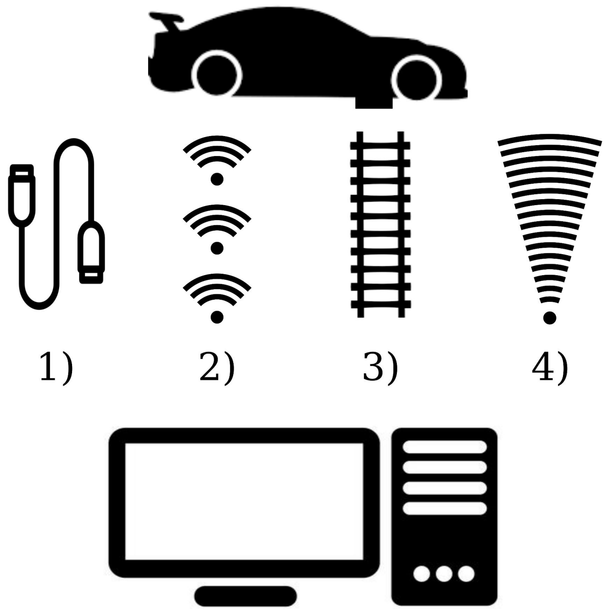
Revision 3 - ⏰

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



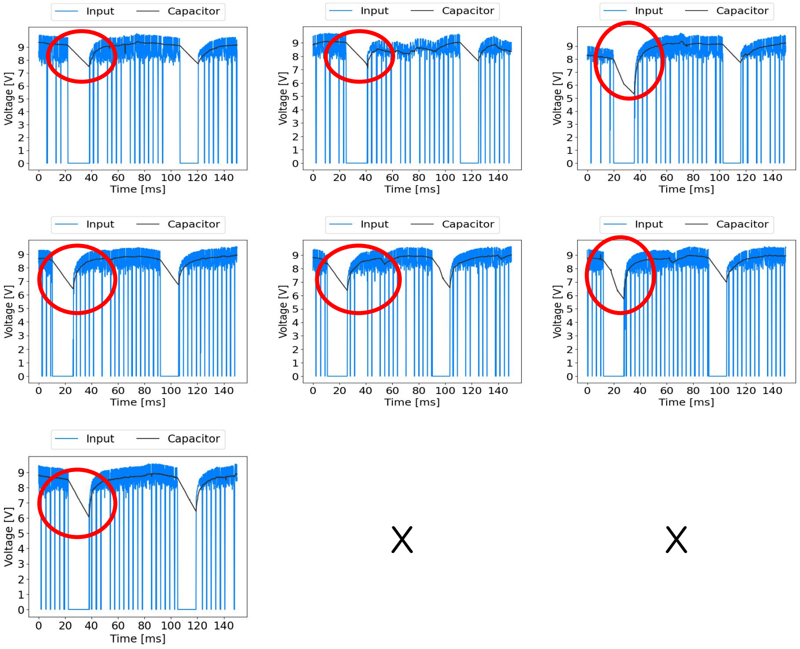
Energy Consumption

No WiFi | Not Sending | Sending

80 MHz

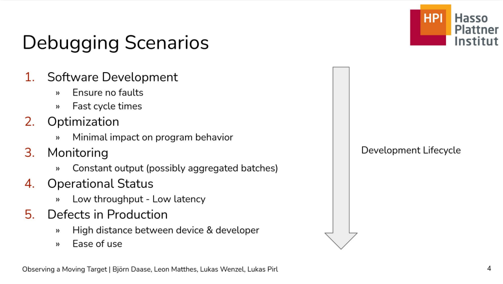
160 MHz

240 MHz



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

Conclusion

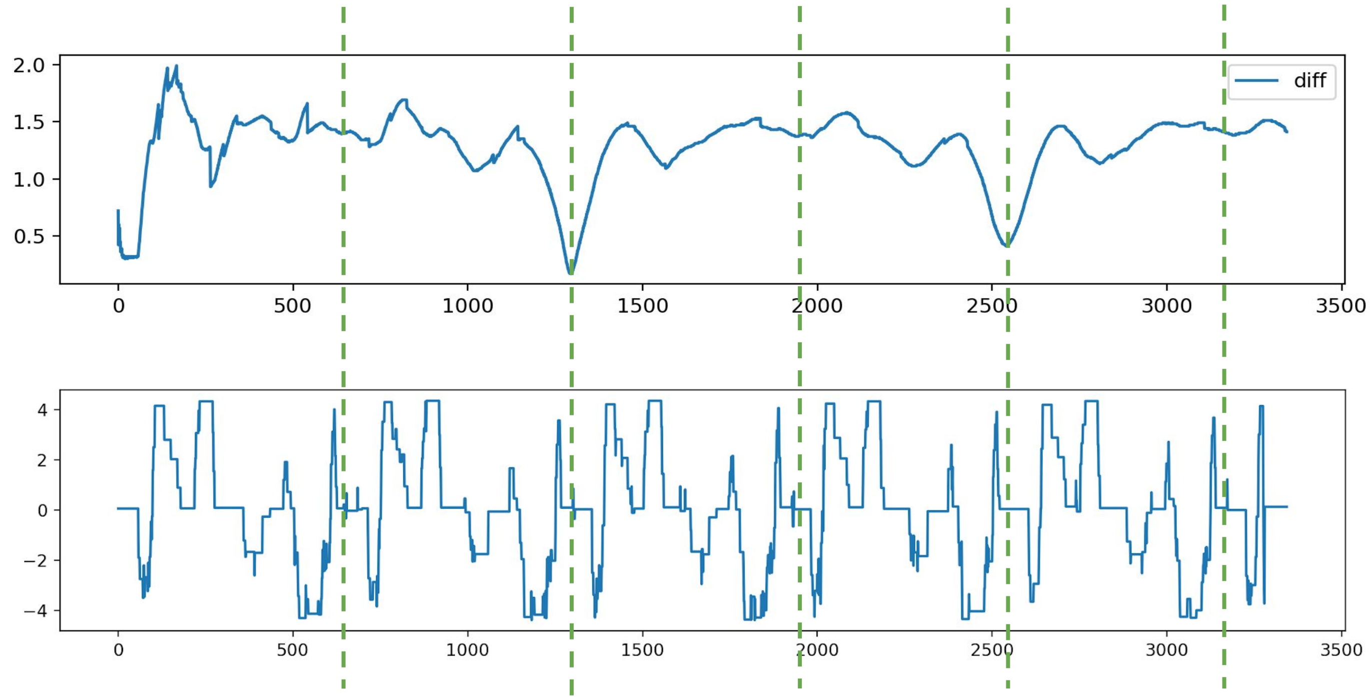


» Wireless debugging
  + offers more applicability & convenience
  - Increased efforts for energy budgeting under constrained circumstances

» Available hard- & software modules require more adaptive instrumentation

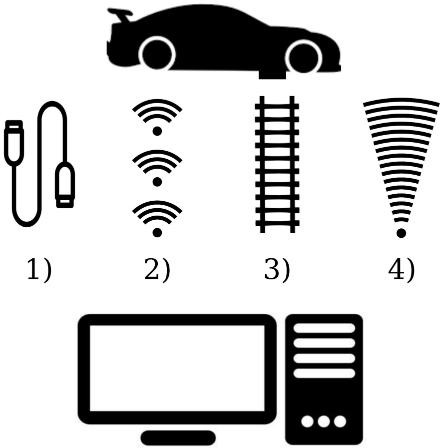
» Automatically select power states based on a configurable energy budget in the future

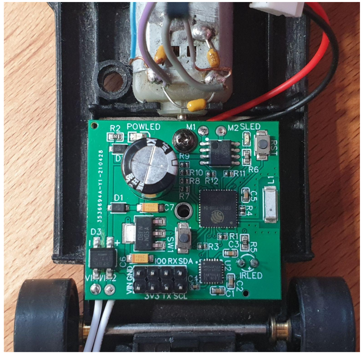
Track Detection

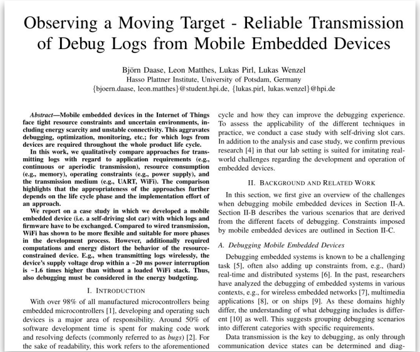


Demo

Video







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

***Abstract*—Mobile embedded devices in the Internet of Things face tight resource constraints and uncertain environments, including energy scarcity and unstable connectivity. This aggravates debugging, optimization, monitoring, etc.; for which logs from devices are required throughout the whole product life cycle.**

In this work, we qualitatively compare approaches for transmitting logs with regard to application requirements (e.g., continuous or aperiodic transmission), resource consumption (e.g., memory), operating constraints (e.g., power supply), and the transmission medium (e.g., UART, WiFi). The comparison highlights that the appropriateness of the approaches further depends on the life cycle phase and the implementation effort of an approach.

We report on a case study in which we developed a mobile embedded device (i.e. a self-driving slot car) with which logs and firmware have to be exchanged. Compared to wired transmission, WiFi has shown to be more flexible and suitable for more phases in the development process. However, additionally required computations and energy distort the behavior of the resource-constrained device. E.g., when transmitting logs wirelessly, the device's supply voltage drop within a ~20 ms power interruption is ~1.6 times higher than without a loaded WiFi stack. Thus, also debugging must be considered in the energy budgeting.

I. INTRODUCTION

With over 98% of all manufactured microcontrollers being embedded microcontrollers [1], developing and operating such devices is a major area of responsibility. Around 50% of software development time is spent for making code work and resolving defects (commonly referred to as *bugs*) [2]. For the sake of readability, this work refers to the aforementioned cycle and how they can improve the debugging experience. To assess the applicability of the different techniques in practice, we conduct a case study with self-driving slot cars. In addition to the analysis and case study, we confirm previous research [4] in that our lab setting is suited for imitating real-world challenges regarding the development and operation of embedded devices.

II. BACKGROUND AND RELATED WORK

In this section, we first give an overview of the challenges when debugging mobile embedded devices in Section II-A. Section II-B describes the various scenarios that are derived from the different facets of debugging. Constraints imposed by mobile embedded devices are outlined in Section II-C.

A. Debugging Mobile Embedded Devices

Debugging embedded systems is known to be a challenging task [5], often also adding up constraints from, e.g., (hard) real-time and distributed systems [6]. In the past, researchers have analyzed the debugging of embedded systems in various contexts, e.g., for wireless embedded networks [7], multimedia applications [8], or on ships [9]. As these domains highly differ, the understanding of what debugging includes is different [10] as well. This suggests grouping debugging scenarios into different categories with specific requirements.

Data transmission is the key to debugging, as only through communication device states can be determined and diag-

Questions?

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

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