

- Das Eisenbahn-Betriebs- und Experimentierfeld (EBUef) der TUB bietet Forschung und Lehre mit verschiedenen Stellwerkstechniken in Verbindung mit einer Modelleisenbahn
- Die Elemente sind remote steuerbar
- Projektideen:
 - ETCS Moving Blocks
 - Eulynx-Stellwerk
 - Co-Simulation mit Modelleisenbahnen
 - Autonomous Train Driving
- Mehr Informationen zum EBUef auf <https://www.ebuef.de/>
- Ansprechpartner: Arne Boockmeyer, Heiko Herholz (EBUef)



Embedded Operating Systems

Operating Systems
and Middleware

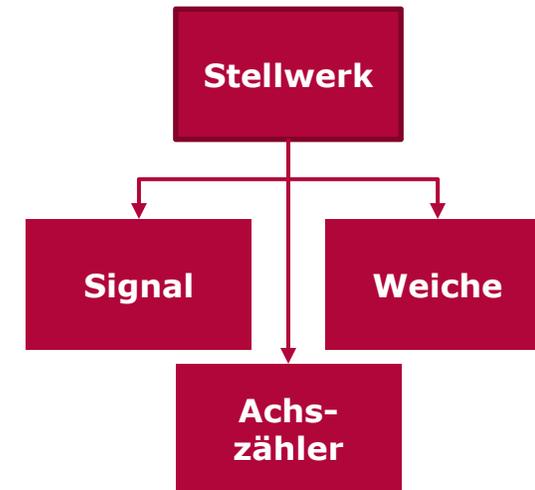
Chart **1**

EULYNX Hardware-in-the-Loop

- Zusammenspiel von EULYNX-Object Controllern im Labor testen
 - Dabei: Simulation eines vollständigen Bahn-Szenarios



- Testszenario in SUMO aufbauen
 - Fahrender Zug, mehrere Gleisabschnitte, Stellwerksbereiche
- Stellwerkslogik implementieren und an die Hardware anbinden
- Hardware automatisieren (Initialisierung, Eingaben auslösen)
- Ansprechpartner: Arne Boockmeyer, Lukas Pirl, Robert Schmid



Embedded Operating Systems
Operating Systems and Middleware

Automatic Train Operation / Automatisches Rangieren

- Einzelwagenverkehr im Güterverkehr nimmt zu:
Rangierbahnhöfe müssen effizienter werden
- Routenplanung, Fahrstraße anfordern,
Kommunikation mit dem Stellwerk
→ Fragen des Bahnbetriebs
- Anbindung an sichere Fahrzeugsteuerungs-Schnittstelle
 - Beachtung der Signale, Meldung der Zugposition,
Berechnung von Bremskurven schon implementiert
- Zielplattform: Oberleitungsrevisionstriebwagen DB 708
- Ansprechpartner: Robert Schmid, Jakob Gärtner (Railergy), DB Netz



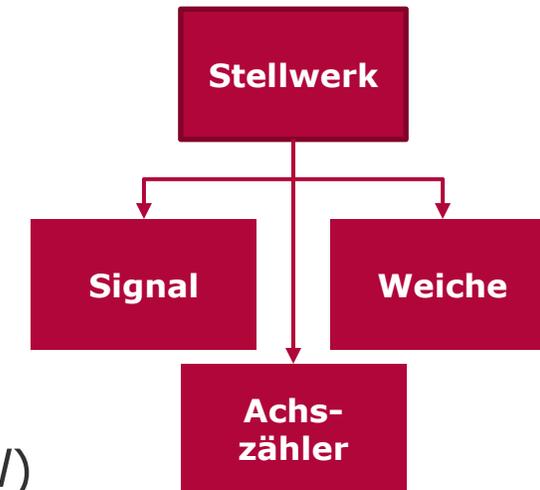
Embedded Operating Systems

Operating Systems
and Middleware

Chart **3**

SIL 4-Software-Entwicklungsprozess am Beispiel RaSTA

- Das "Rail Safe Transport Application"-Protokoll (RaSTA) bildet die sichere Transportschicht im IP-Protokollstack zwischen den EULYNX-Stellwerkskomponenten
 - Idee: Zwei redundante Ethernetkabel
- Offene Referenzimplementierung erleichtert Hardwareanbietern den Markteintritt und kann für Systemtests verwendet werden
 - Muss "SIL-4" Sicherheits-Anforderungen genügen (*safety integrity level*)
 - Offene Fragen in der aktuellen Spezifikation des Protokolls
- Ansprechpartner: Robert Schmid, Stefanos Moumouris (SBB), Ibtihel Cherif (DB Netz)

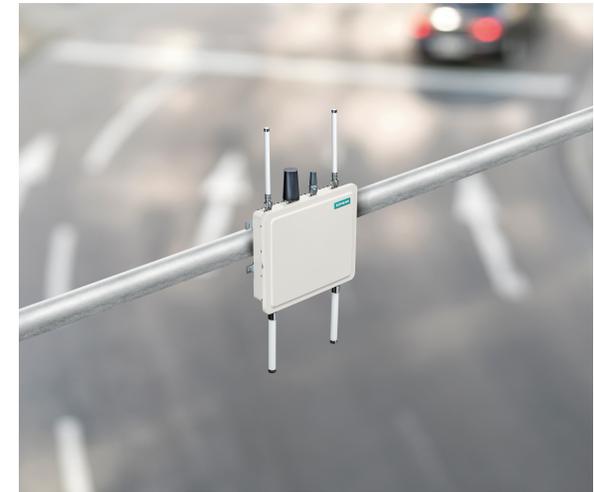


Embedded Operating Systems

Operating Systems and Middleware

Chart 4

- Kontext: Betrieb und Wartung von Geräten ohne physischen Zugang
 - z. B. V2X-Knoten an Ampeln, mobile IoT-Geräte
- Betrieb
 - Test-Suite mit üblichen Fehlerursachen (Fehlerinjektion)
 - Gibt es sie? Wenn nein, wie sähe sie aus?
 - Überlast (z. B. OOM, RT-Livelock, soft/hard lockups)
 - Günstiger DIY-Hardware-Watchdog
 - Wie könnte so etwas gebaut werden?
- Wartung
 - Wiederherstellen eines "last known good" Zustands (Konf., Kernel, ...)
 - z. B. bei fehlgeschlagenen Aktualisierungen



Embedded Operating Systems

Operating Systems and Middleware

Chart 5

■ Tastaturen und -Mäuse:

- Snooping, Replay, Spoofing
- Was wird im HPI gefunkt (ISO 27001)?
- Vorarbeit existiert



– d. h. Recherche um neuartiges Ziel für Projekt ausmachen notwendig

■ Funksteckdosen

- Wir haben ein Modell mit noch nicht verstandenem Protokoll...
- Steuerung mittels GPIO (Arduino, Single-Board Computer, etc.)
- Anwendung: Automation von HW-Testbeds, Fehlerinjektion, ...



**Embedded
Operating Systems**

Operating Systems
and Middleware

Chart **6**

■ Ansprechpartner: Lukas Pirl

- Inspiriert von RailChain
 - im Zug verteilte Blockchain auf COTS-Hardware als Blackbox
- Fragen, wie: *Können Sensor-Knoten selbst teilnehmen?*
 - Im Fokus: Zeitverhalten
- Vermessung
 - (Quasi-)Standard-Benchmarks
 - Gibt es sie?
 - Wenn nicht, wie sähen sie aus?
- Automation von Hardware-Testbeds
 - Ausfälle und Störung von Netzwerk (inkl. Wi-Fi), Strom, Last, etc.
- Ansprechpartner: Lukas Pirl, Robert Schmid



Embedded Operating Systems

Operating Systems
and Middleware

Chart 7