

Projektseminar Echtzeitsysteme

Endvortrag



TECHNISCHE
UNIVERSITÄT
DARMSTADT

**Team Nullpointer: Ralf Kundel, Marcel Mann,
Alina Weber, Robert Wiesner**

```
Exception in thread "main" java.lang.NullPointerException  
    at Fahrer.aufwecken(Fahrer.java:832)  
    at Motor.run(Motor.java:7)  
    at Auto.drive(Auto.java:42)  
    at Auto.main(Auto.java:13)
```



Roadmap

- Überblick
- Wandfolger
- orthogonales Einparken
- RC-Modus
- C2C
- Probleme
- Zeitplanung



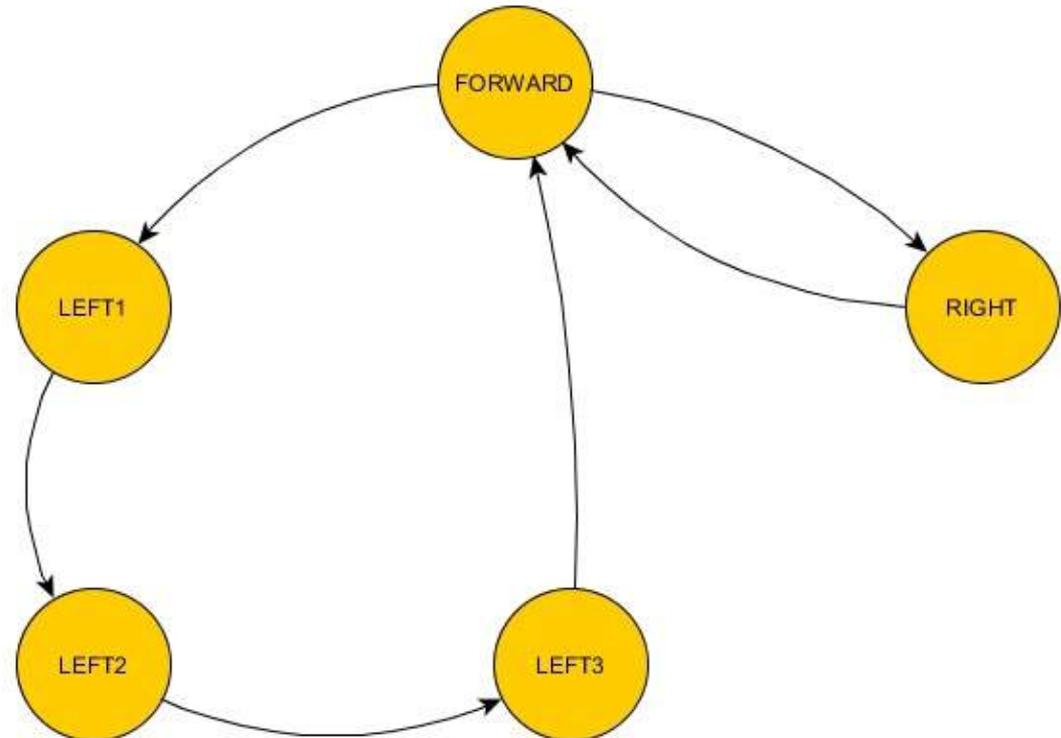
Überblick - Aufgaben

- Wandfolger
 - sehr gute Regelung (maximal Speed)
 - Rechts- und Linkskurven
- RC-Modus
 - Fernsteuerung via PC und Handy
 - Modus durch Handy wechseln
- Parken
 - paralleles Einparken
 - rückwärts Einparken
- Car-2-Car Kommunikation
 - „funktioniert“ mit 2 (und 3) Autos

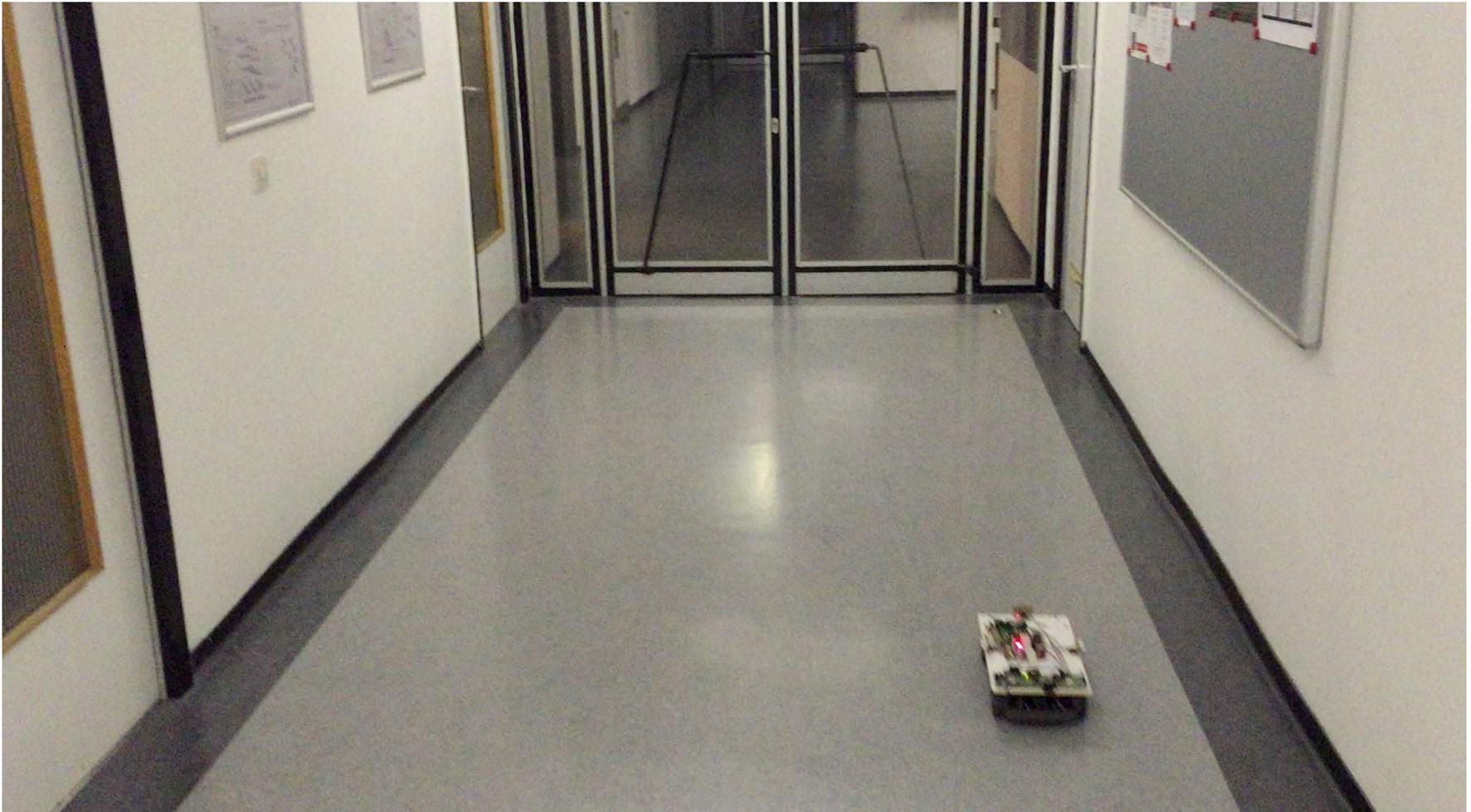


Wandfolger

- Geradeausfahren
 - PD-Regler
 - Deltaberechnung in API
 - API: Servo-Offset
 - Türrahmenfilter
- Erweiterung Linkskurven
 - Frontdistanz
 - Rücksetzen
- Erweiterung Tempomat
 - problematisch



Demovideo: Linkskurve

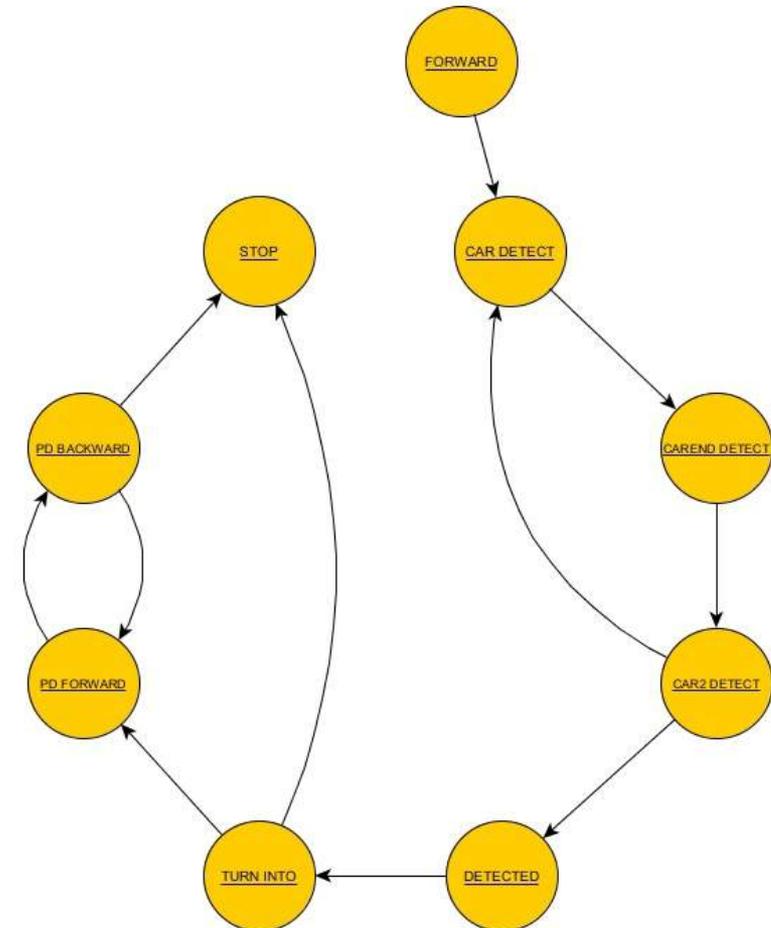


Demovideo: Speedrunde



orthogonales Parken

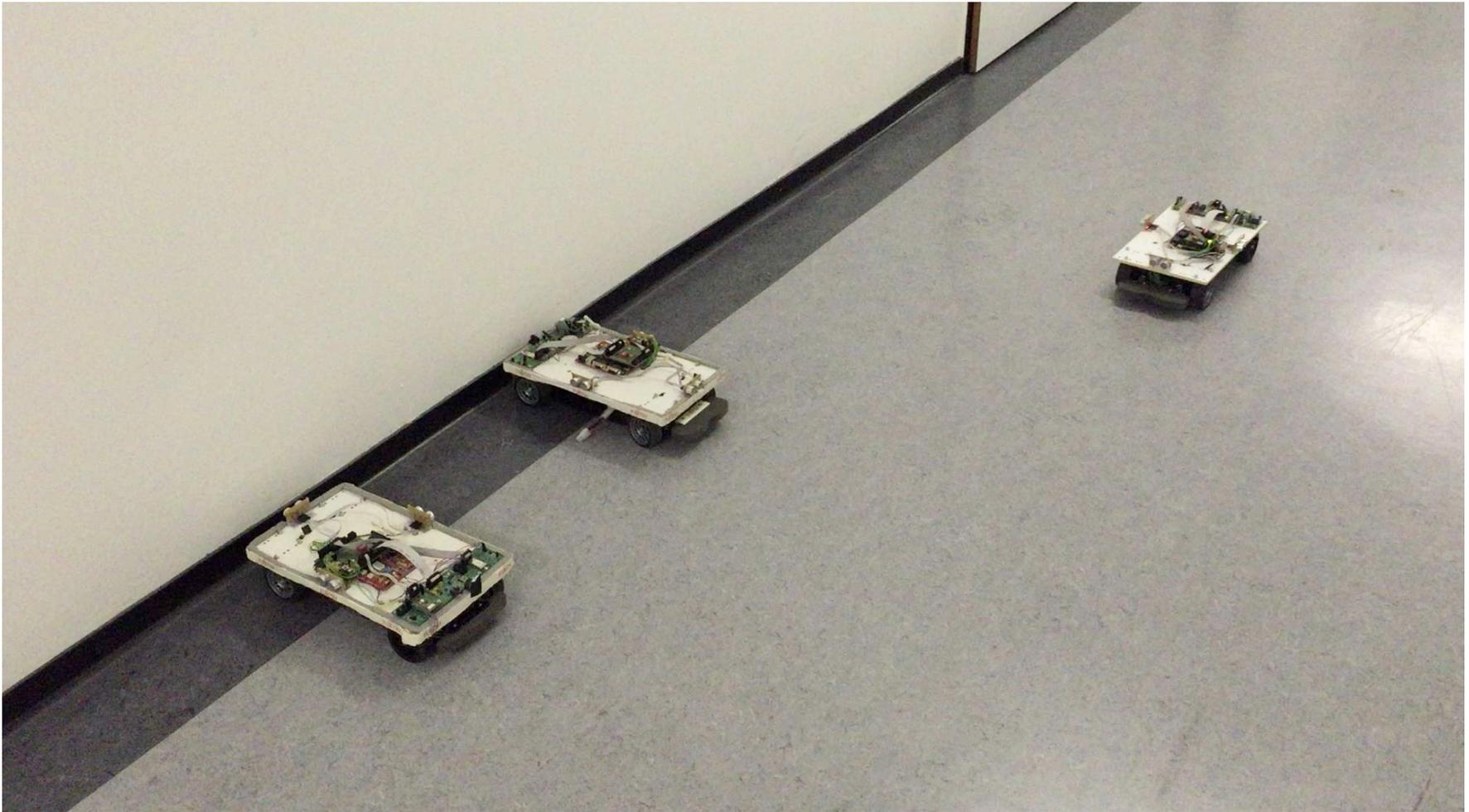
- Erkennen der Parklücke
- „relativ“ statisches Einparken
- Herausforderungen:
 - kein Sensor hinten
 - PD-Regler benötigt ebene Wand
 - Autos nicht baugleich



Demo: orthogonales Einparken



TECHNISCHE
UNIVERSITÄT
DARMSTADT



- Smartphone Bluetooth Fernsteuerung
 - „Mode-Picker“
 - z.B.: Umschalten von automatisiertem Fahren zu manuellem Fahren
 - ferngesteuertes Fahren
 - Debugging
 - Parameter für PD-Regler setzen
 - Sicherheitszone („Assistant Drive“)
- PC Fernsteuerung (Sticks)
 - geringere Latenz als Bluetooth
 - Steuerung mit Xbox Controller



Beispielcode: Handy App

```
when BT_Connect_Button .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .Disconnect
    set BT_Status_Label .Text to "Disconnected!"
    set BT_Connect_Button .Text to "Connect"
    set CheckBoxModes .Enabled to false
    set CheckBoxModes .Enabled to false
    set CheckBoxModes .Enabled to false
  else
    if not BluetoothClient1 .Enabled
    then
      call Notifier1 .LogWarning
      message "Please enable Bluetooth"
    if call BluetoothClient1 .Connect
      address BT_Dev_Picker .Selection
    then
      set BT_Status_Label .Text to "Connected!"
      set BT_Connect_Button .Text to "Disconnect"
      set CheckBoxModes .Enabled to true
      set CheckBoxRC .Enabled to true
      set CheckBoxC2C .Enabled to true
    else
      set BT_Status_Label .Text to "Connection Failed!"
```

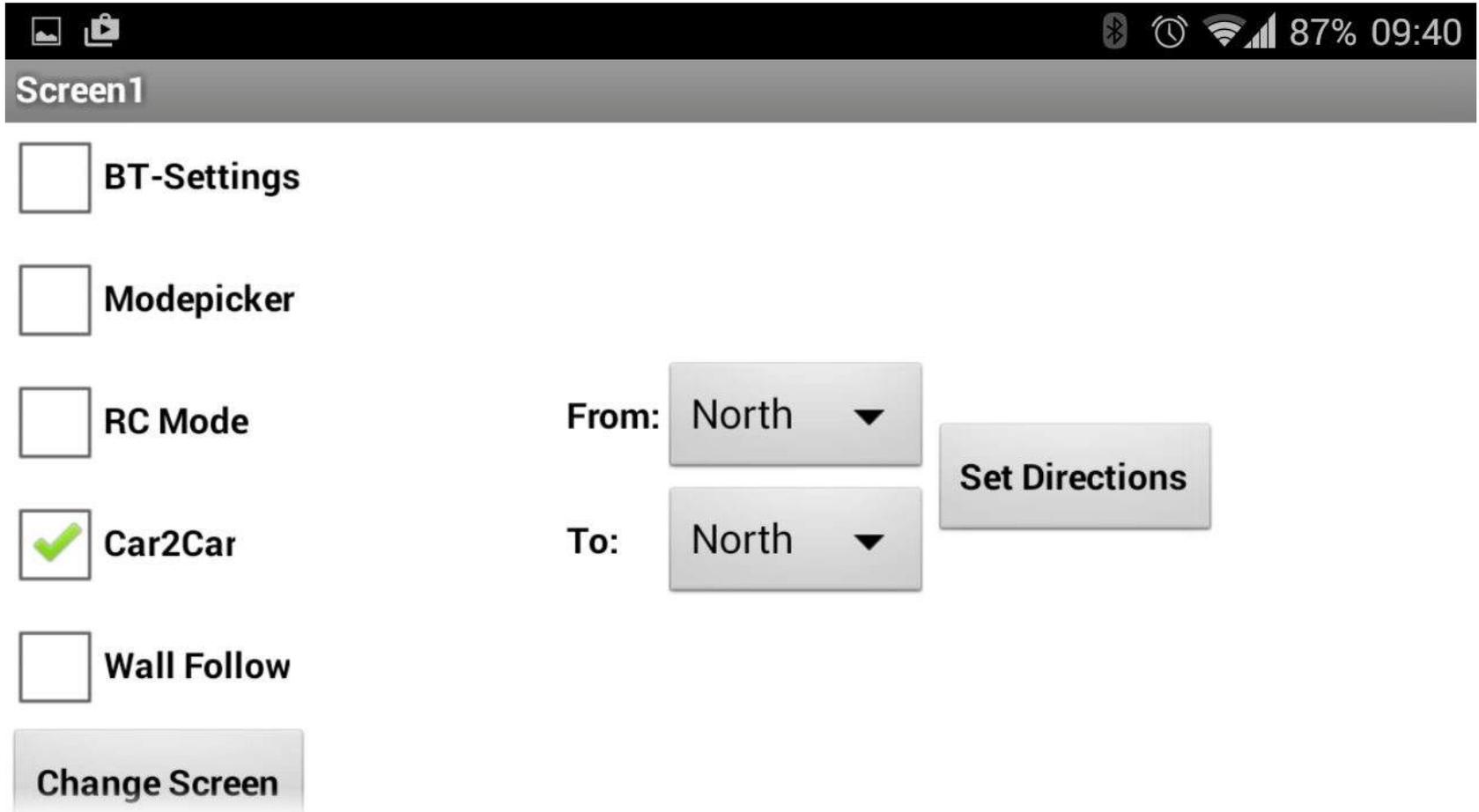
```
when ReconnectButton .Click
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .Disconnect
    set BT_Status_Label .Text to "Disconnected!"
  if call BluetoothClient1 .Connect
    address BT_Dev_Picker .Selection
  then
    set BT_Status_Label .Text to "Connected!"
    set Clock1 .TimerEnabled to true
    set BT_Connect_Button .Text to "Disconnect"
  else
    set BT_Status_Label .Text to "Connection Failed!"

when BT_Dev_Picker .BeforePicking
do
  set BT_Dev_Picker .Elements to BluetoothClient1 .AddressesAndNames

when BT_Dev_Picker .AfterPicking
do
  set BT_Dev_Picker .Text to segment text BT_Dev_Picker .Selection
  start 19
  length 7
```



die App!



Screen1

BT-Settings

Modepicker

RC Mode

Car2Car

Wall Follow

From: North ▼

To: North ▼

Set Directions

Change Screen

The screenshot shows an Android application interface. At the top, there is a status bar with icons for camera, clipboard, Bluetooth, alarm, Wi-Fi, and cellular signal, along with the battery level at 87% and the time 09:40. Below the status bar is a grey header labeled 'Screen1'. The main content area contains a list of settings, each with a checkbox: 'BT-Settings', 'Modepicker', 'RC Mode', 'Car2Car' (which is checked with a green checkmark), and 'Wall Follow'. To the right of these settings are two dropdown menus labeled 'From:' and 'To:', both currently set to 'North'. A 'Set Directions' button is positioned to the right of these dropdowns. At the bottom left, there is a 'Change Screen' button.



Car-2-Car

- Implementierung von „Rechts vor Links“
- Fahrzeuge senden:
 - von wo sie kommen
 - wo sie hinfahren wollen
- Einfaches Protokoll mit Acknowledge Paketen
- dezentrale Auswertung der Verkehrssituation
- theoretisch bis zu vier Autos



Demovideo: Car-2-Car



neue ungelöste/unlösbare Probleme

- defekte Hardware
 - Liniensensoren
 - US-Sensoren
- schwankende Prozessoren
 - deutlich merkbare Leistungsunterschiede
 - Abstürze bei Empfang von Nachrichten (Auto 1)
- Funkmodule unzuverlässig



Auto 6: keine Hardware Bugs



Zeitplanung

<http://www.123gif.de/gifs/weihnachtsmann/weihnachtsmann-0068.gif>



Anfang
Nov.
2014

Anfang
Dez.
2014

Anfang
Feb.
2015

Ende
Feb.
2015

geradeaus fahren

geradeaus fahren

Kurven

Kurven

Car-2-Car

Einparken

o. Einparken

Rc-Modus

RC-Features

kooperative
Automation



Zeitbedarf

<i>Tätigkeit</i>	<i>geplant</i>	<i>Tatsächlich</i>
<i>PD-Regler</i>	<i>20</i>	<i>150</i>
<i>Kurven</i>	<i>50</i>	<i>100</i>
<i>RC-Modus</i>	<i>70</i>	<i>60</i>
<i>Einparken</i>	<i>70</i>	<i>30</i>
<i>orth. Einparken</i>	<i>50</i>	<i>20</i>
<i>C2C</i>	<i>80</i>	<i>150</i>
<i>Optimierungen</i>	<i>0</i>	<i>300</i>
		<i>750</i>

Zeit in Personenstunden



Fazit

- Spaß
- deutliche Zeitverluste durch Hardware-Bugs
- bessere Hardware (z.B. Raspberry Pi)
- positiv:
 - freie Zeiteinteilung
 - praktische Aufgabenstellung





Alina Weber

alina_sophie.weber@stud.tu-darmstadt.de

Marcel Mann

marcel.mann@stud.tu-darmstadt.de

Ralf Kundel

ralf.kundel@stud.tu-darmstadt.de

Robert Wiesner

robert.wiesner@stud.tu-darmstadt.de

