



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Fakultät Elektrotechnik und Informationstechnik Institut für Automatisierungstechnik

Belegverteidigung - Entwurf eingebetteter Systeme

Tobias Steege und Sascha Steffen

Dresden, 2013-07-17



**DRESDEN
concept**
Exzellenz aus
Wissenschaft
und Kultur

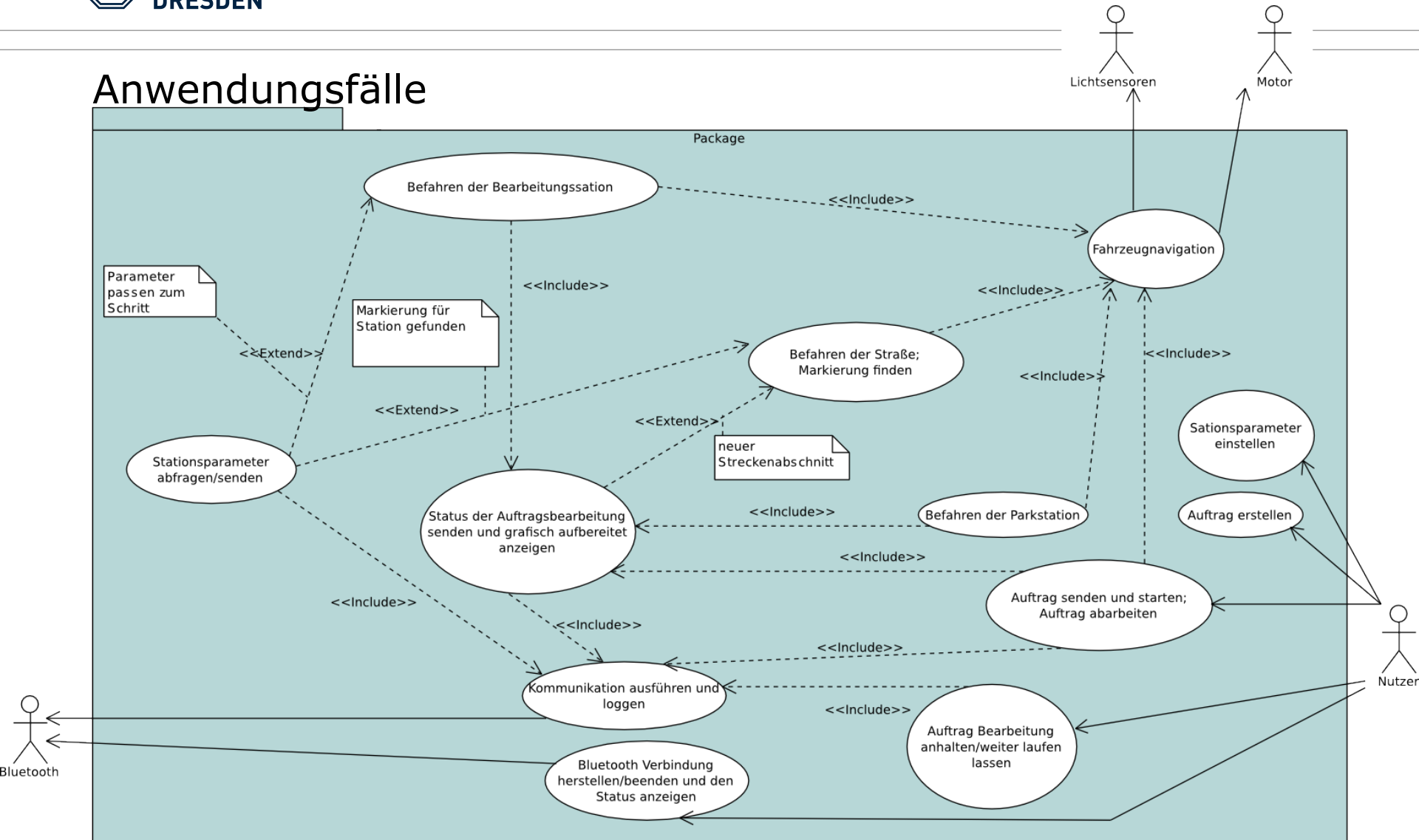
Inhalt

- Aufgabe
- Anforderungen
- Entwurf
 - NXT
 - RasPi
 - Kommunikationsprotokoll
- Implementierung
- Auswertung / Verbesserungen

01 Aufgabe

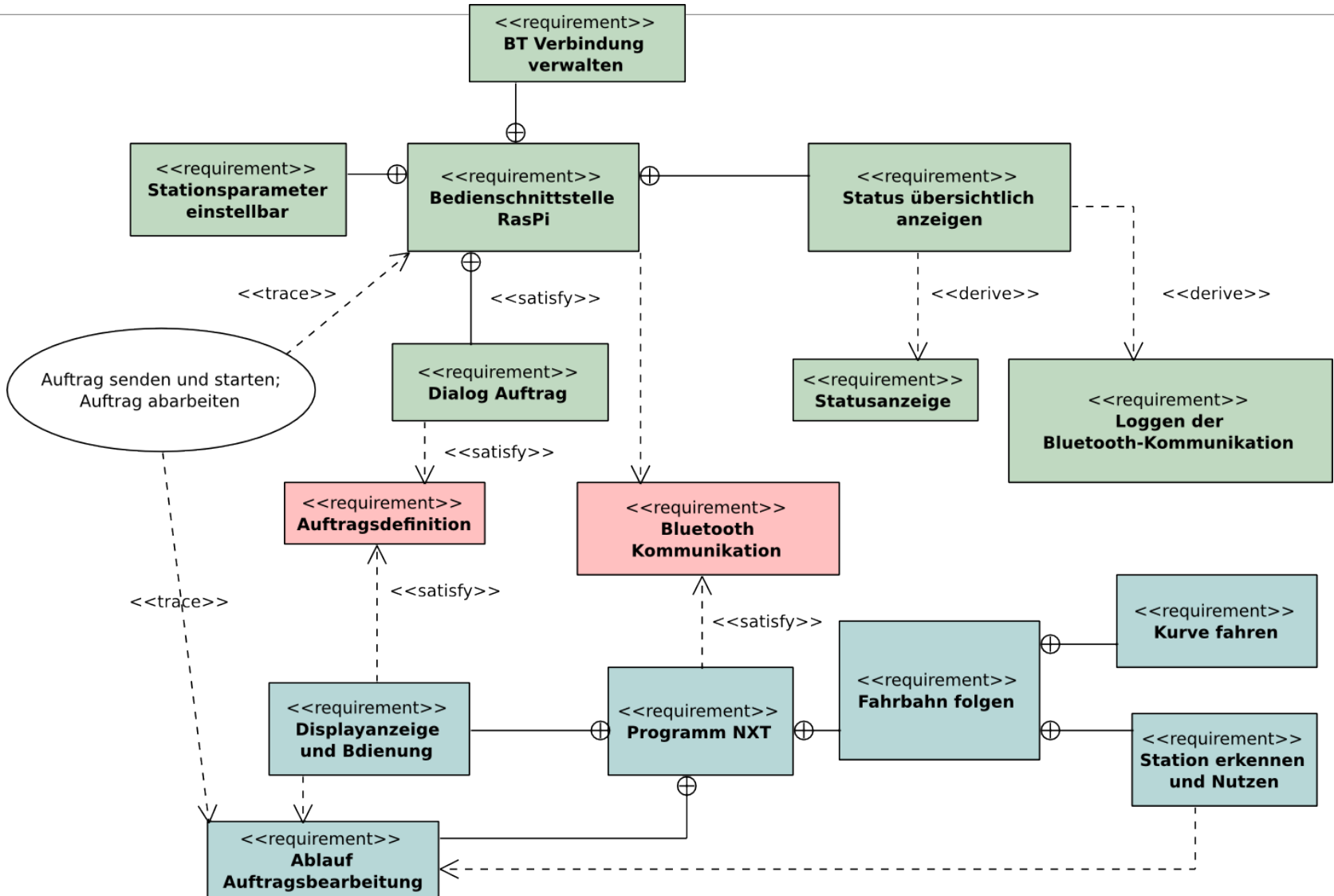
- Fahrbahn
 - Parkstelle als Ausgangspunkt
 - 4 Buchten symbolisieren „Bearbeitungsstationen“
- 1 NXT fährt auf Bahn und „bearbeitet“ Aufträge
 - Sensorauswahl
 - Fahrzeug-„Design“
- Interaktion mit RaspberryPi (über Bluetooth)
 - Annahme von Auftragsdaten
 - Senden von Statusdaten
 - Abfragen von Qualitätsdaten der „Bearbeitungsstationen“
 - Start / Stopp Kommando
- Nutzeroberfläche auf RaspberryPi
 - Bluetooth-Verbindung herstellen
 - Auftrag definieren
 - Qualitätsparameter von Bearbeitungsstationen festlegen

Anwendungsfälle



02 Anforderungen

Requirements-Diagramm



03 Entwurf

Varianten - NXT

Fahrzeugdesign

- größtenteils Vorgegeben
- Schwerpunkt haben wir nach unten Verlegt
- Kabelführung verändert, dadurch Übersichtlicher

Sensoren

- Nutzung von zwei Lichtsensoren

03 Entwurf

Varianten - NXT

Programm

Variante 1:

- Folgen über beide Sensoren, dabei immer auf schwarz bleiben
- erkennen von Kurven, wenn beide Sensoren weiß sind
- erkennen von Stationen, wenn Sensoren Grau anzeigen
- Unterscheidung der Stationen über RFID Tag

Vorteile: - NXT kann an jeder Stelle auf der Bahn anfangen
- Sollte der NXT sich verfahren, kann er einfach wieder auf die Bahn gesetzt werden

Probleme: - Unterschied zwischen Grau und Weiß sehr gering
- Manchmal werden Kanten als Grau erkannt
- RFID Sensor hat geringe Reichweite

03 Entwurf

Varianten - NXT

Variante 2: gewählte Variante

- Folgen der Bahn mittels der beiden Sensoren
- Nur Erkennung ob die Bahn verlassen wurde (egal ob Weis an Kurve oder auf ein graus Feld gefahren).
- NXT entscheidet anhand seiner Position, ob er um die Kurve fährt oder auf einem grauen Feld steht

Vorteile:

- Unterscheidung zwischen Weis und Grau fällt weg
- NXT kennt seine Position -> kann an RasPi geschickt werden
- Stationen werden nicht anhand von RFID-Tags unterschieden

Probleme:

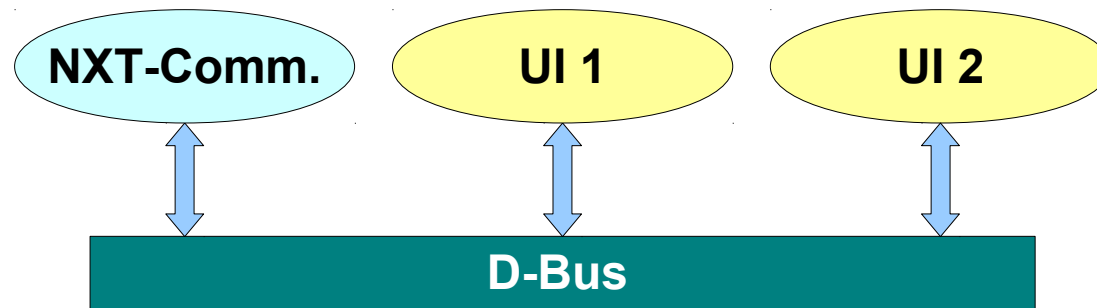
- NXT muss immer an der selben Stelle starten
- wenn der NXT sich verfährt, kann er nicht einfach wieder auf die Bahn gesetzt werden
- wenn die Fahrbahn geändert wird, z.B. durch weniger Station muss auch der Quellcode geändert werden

03 Entwurf

Varianten - RaspberryPi

Variante 1:

- Kommunikation mit Fahrzeug und Nutzerschnittstelle trennen
⇒ prinzipiell mehrere Nutzerschnittstellen denkbar (grafisch, Konsole, Web...)
- Problem: Schnittstelle zwischen den 2 Anwendungen nötig
⇒ DBus:
 - Standard-Bestandteil fast aller Linux-Distros
 - perfekt für standardisierte Anwendungsschnittstellen (Bsp. Skype, MPRIS¹)



03 Entwurf

Varianten - RaspberryPi

- Variante 1 würde weitere Schnittstellendefinition nötig machen
- konkrete Anwendung zu speziell, um von modularem Aufbau wirklich zu profitieren
- Aufwand zu groß

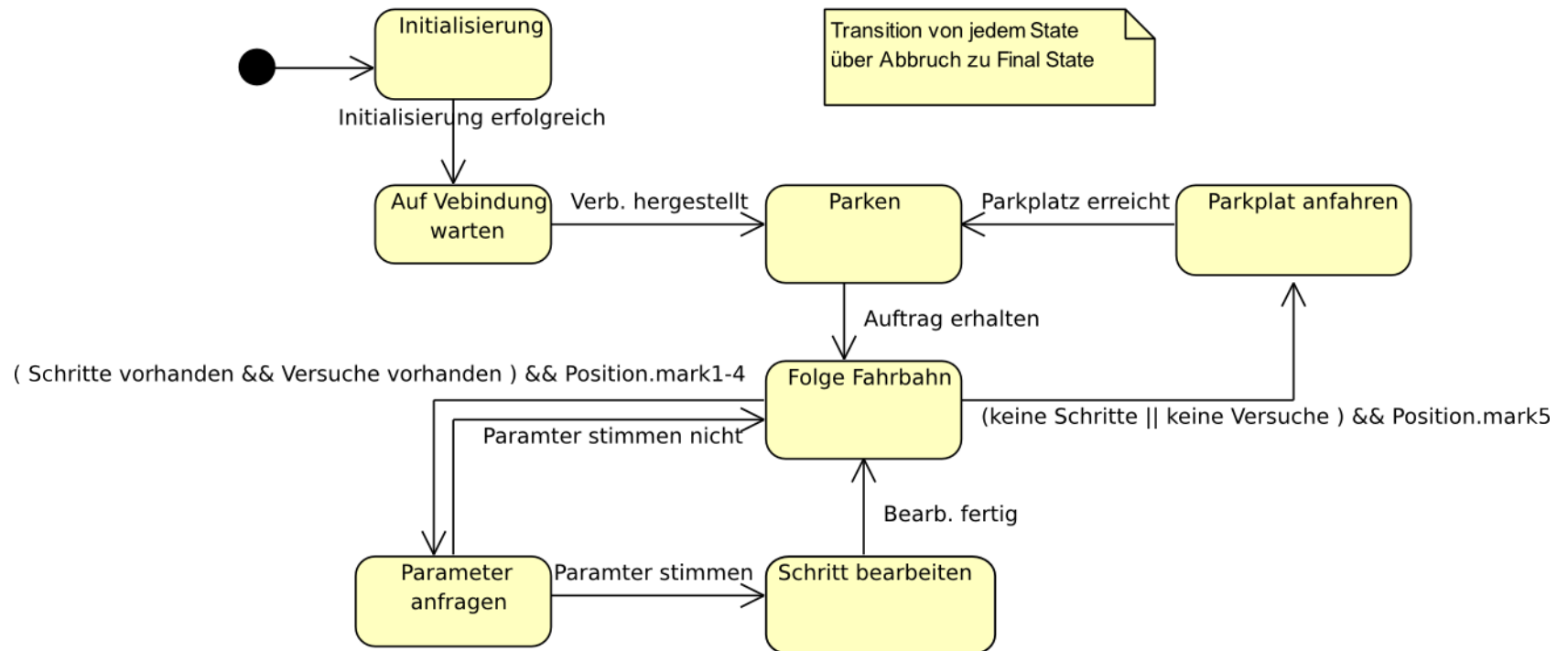
Variante 2:

- GUI Anwendung mit Qt
- komfortabel mit C++ zu schreiben
- Aufwand bei GUI-Design durch grafischen Editor verringert
- übersichtliche Anwendung ohne Konsolenwissen bedienbar
- per ssh und X-Forwarding auch für Fernzugriff geeignet



03 Entwurf

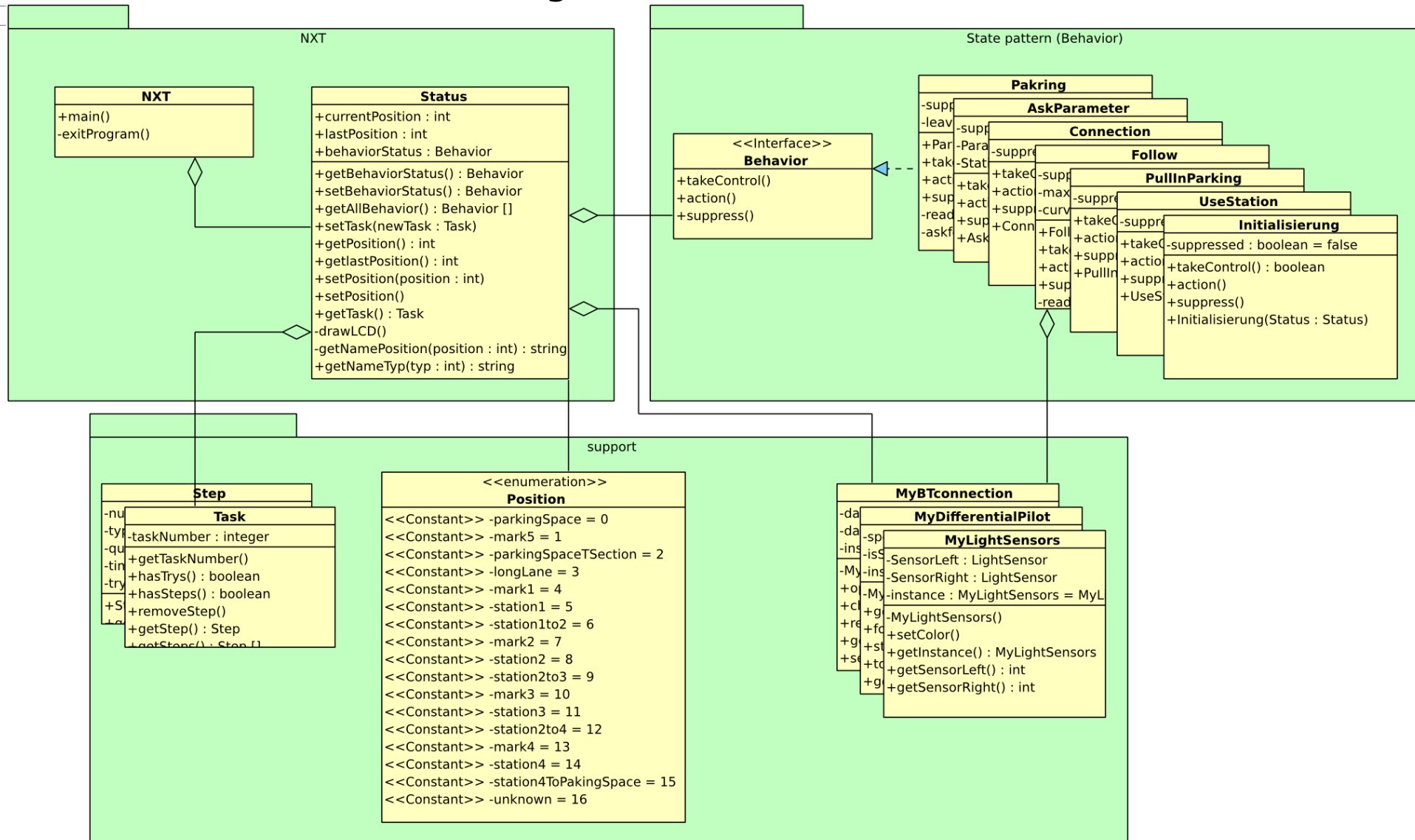
Zustandsmaschine NXT





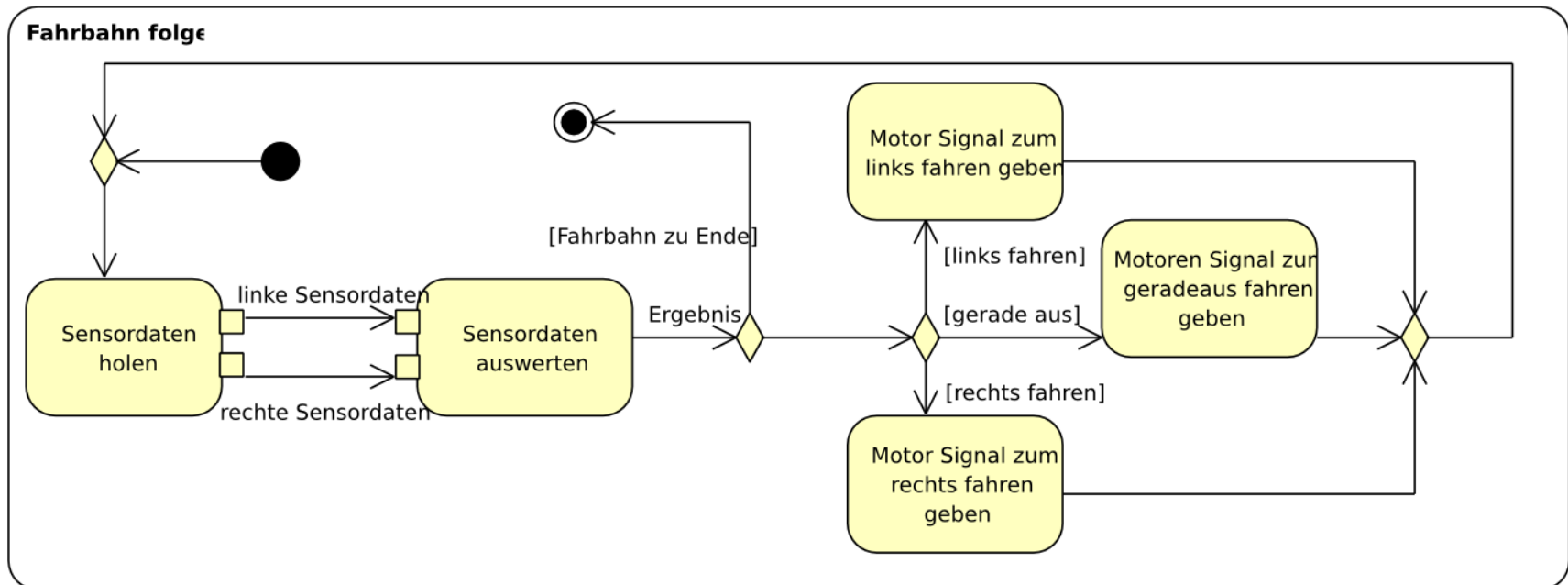
03 Entwurf

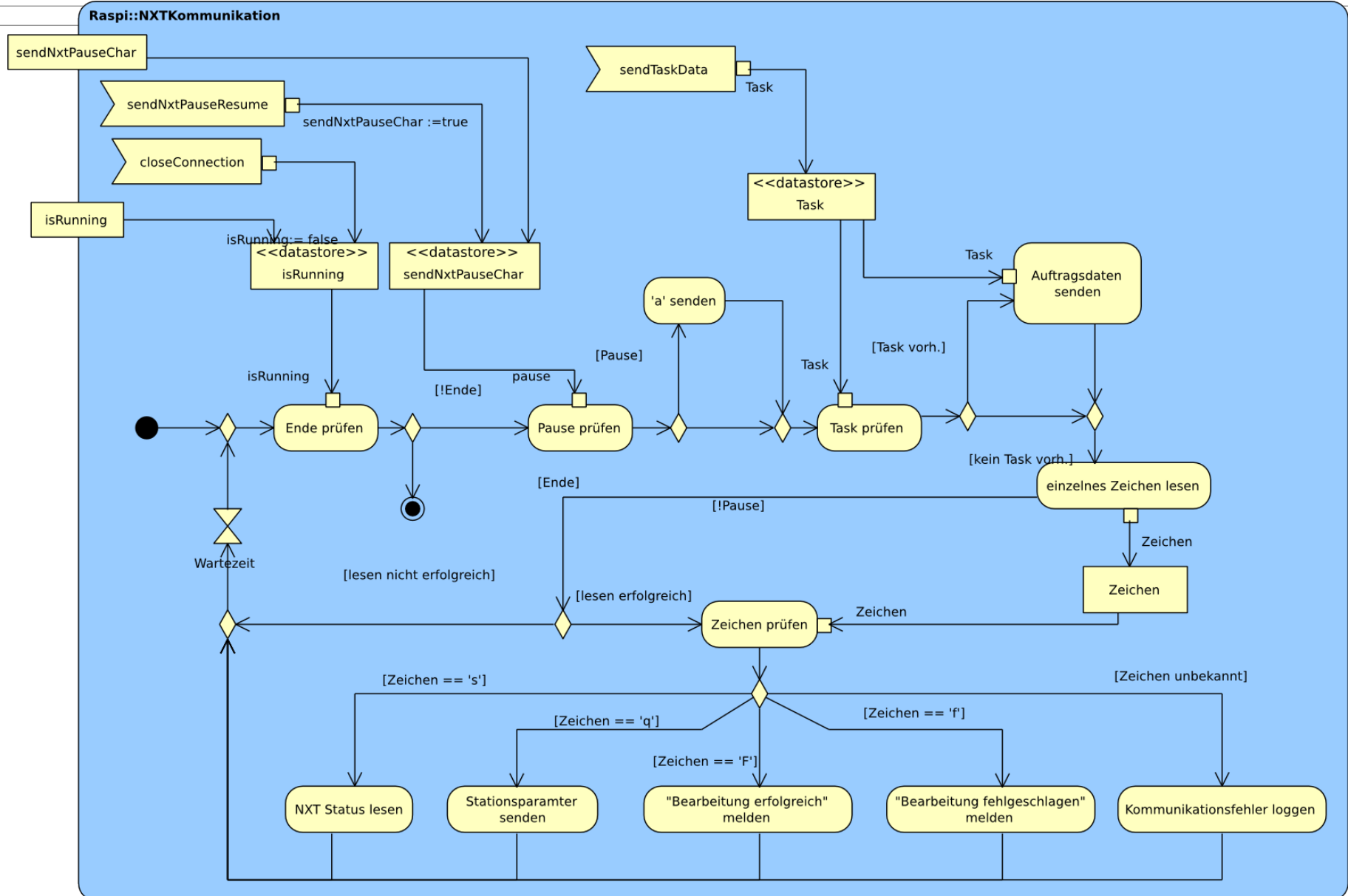
Klassendiagramm NXT



03 Entwurf

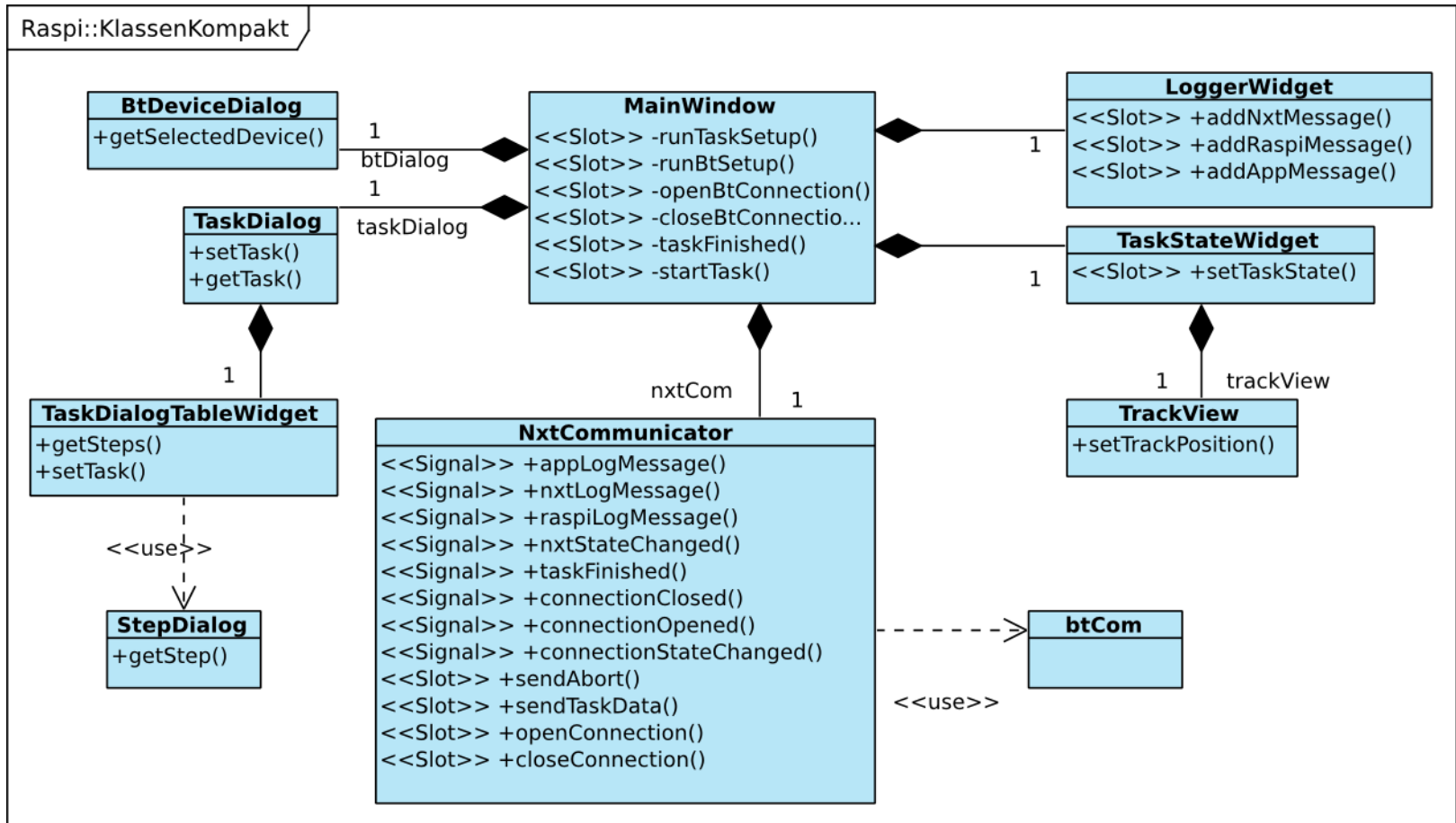
Aktivität – Fahrbahn folgen





03 Entwurf

Klassendiagramm RaspberryPi



04 Entwurf

Kommunikationsprotokoll

- Start mit Buchstabe (char)
→ Eindeutige Unterscheidung von Nachrichtentyp über erstes Zeichen
- möglichst dateneffizient
- schnelle Verarbeitung auf NXT möglich

Byte-Nr	0	1	2-5	$6+i*3$	$7+i*3$	$8+i*3$
Auftrag	't'=0x74	[Auftr.Nr \in [1,99]]	[Schritte]	[Typ \in [0,2]]	[Quali \in [1,4]]	[Zeit \in [3,20]]

04 Entwurf

Kommunikationsprotokoll

Byte-Nr	0	1	2-5
Statusnachricht	's'=0x73	[Pos. \in [0,15]]	[Schritt Nr.]

Byte-Nr	0
Auftrag erfolgreich	'F'=0x46
Auftrag fehlgeschlagen	'f'=0x66
Pause/Weiter	'a'=0x61

04 Entwurf

Kommunikationsprotokoll

Qualitätsanfrage

Byte-Nr	0	1
Anfrage Quali	'q'=0x71	[Station \in [1,4]]

passende Antwort

Byte-Nr	0	1	2
Antwort Quali	'q'=0x71	[Typ \in [0,2]]	[Quali \in [1,4]]

Protokoll-Log

```
(00:16:53:08:12:ac)
23:08:04 [App ] rfcomm-Befehl Fehler: Xlib: extension "RANDR" missing on
display ":11.0".

23:08:09 [App ] serielle schnittstelle geöffnet.
23:09:00 [RasPi] Task:
t 0x02 0x02 0x00 0x00 0x00
  0x00 0x02 0x03
  0x01 0x03 0x04
23:09:00 [NXT ] s (NXT-Status): 0x00 0x01 0x00 0x00 0x00 (1)
23:09:02 [NXT ] s (NXT-Status): 0x03 0x01 0x00 0x00 0x00 (1)
23:09:04 [RasPi] Pause/Weiter)
23:09:10 [RasPi] Pause/Weiter)
23:09:22 [NXT ] s (NXT-Status): 0x04 0x01 0x00 0x00 0x00 (1)
23:09:22 [NXT ] q 0x01 ( Qualitätsanfrage)
23:09:22 [RasPi] Qualitätsparameter gesendet: q 0x02 0x01
23:09:23 [NXT ] s (NXT-Status): 0x06 0x01 0x00 0x00 0x00 (1)
23:09:25 [NXT ] s (NXT-Status): 0x07 0x01 0x00 0x00 0x00 (1)
23:09:25 [NXT ] q 0x02 ( Qualitätsanfrage)
23:09:25 [RasPi] Qualitätsparameter gesendet: q 0x00 0x02
23:09:25 [NXT ] s (NXT-Status): 0x08 0x01 0x00 0x00 0x00 (1)
23:09:40 [App ] rfcomm-Befehl: Connected /dev/rfcomm0 to 00:16:53:08:12:AC on
channel 1
```

04 Implementierung

RaspberryPi

Zusammenarbeit

- gemeinsames Git-Projekt
- RasPi gut fernsteuerbar
 - unabhängiges (individuelles) Arbeiten möglich
 - gleichzeitig zeitnahes Einarbeiten von Änderungen
- regelmäßige gemeinsame Testläufe

Schwierigkeiten

- Kompilieren auf RasPi: zeitaufwendig, aber: Cross-Compile-Umgebung für Qt ebenfalls zeitaufwendig zu erstellen
- NXT Bluetooth-Komm. schwer zu debuggen

05 Auswertung / Verbesserungen

Protokoll

- Pause/Stopp besser über zwei Zeichen (z.Bsp. 'a', 'A')
- weniger fehleranfällig

genauere Positionsanzeige

- Strecke weiter unterteilen, bzw. Koordinaten
- möglicherweise animierte Vorschau

gesendete Daten verifizieren

- Hash anhängen



Demo :-)

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?