

# EZL - Abschlusspräsentation

---

## NXT Standalone

Sebastian Berumen  
Dominik Elberskirch  
Thomas Windisch



# Inhalt

---

- Überblick
- Anwendungsanalyse
- Komponenten & Schnittstellen
- Komponententest & WCET Bestimmung
- Komposition
- Akzeptanztest
- Kleine Vorführung



# Überblick

---

- LEGO Mindstorm
- Atmel-32-Bit-ARM-Prozessor 256 kB Flash, 64 KB RAM, 48 MHz
- Koprozessor: Atmel 8-Bit AVR, 4 KB Flash, 512 Byte RAM, 8 MHz
- 3 Motorausgänge mit Rückkanal
- 4 Sensoreingänge, analog und digital (I<sup>2</sup>C) kombiniert
- Punktmatrix LCD: 100 × 64 Pixel, Abmessungen: 26 × 40,6 mm
- Sound: 8-Bit, Samplingrate von 2 – 16 kHz

→ Aufbau eines aufrecht rollenden Roboters

# Anwendungsanalyse

---

- Umweltabhängigkeiten
  - Maximale Sensormessfrequenzen
  - Trägheit der Aktuatoren
  - Mechanische Trägheit des Roboters
  - Temperatureinfluss
  - Bodenbeschaffenheit
  - Stromversorgung des Systems
  - Hindernisse

# Anwendungsanalyse

---

Ereignisse	Ergebnisse
Sensormesswerte	Aktuatorenstellwerte
Stellen d. Aktuatoren	Lageänderung des Systems
Lageänderung	Korrektur

# Anwendungsanalyse

---

Ziel:

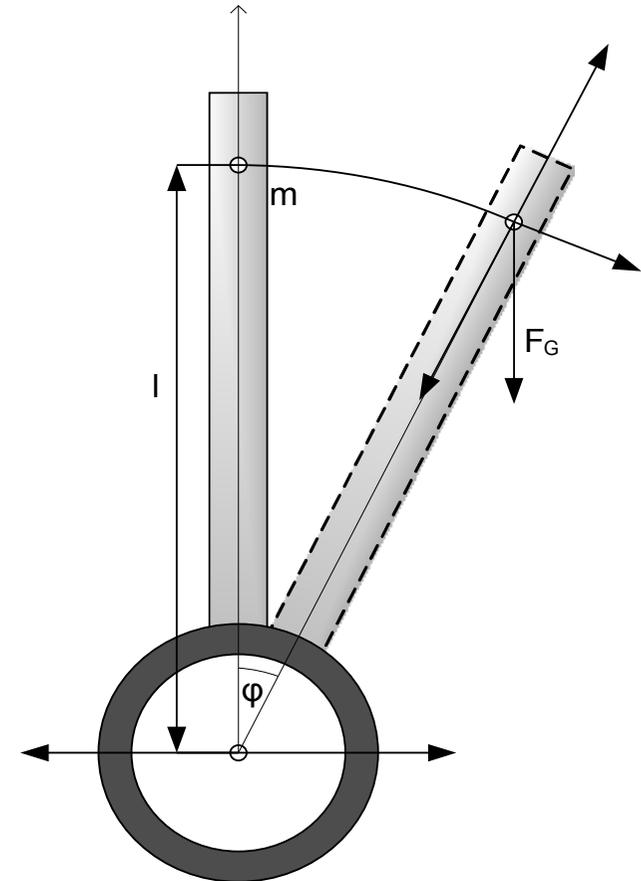
- NXT Standalone soll nicht umkippen
- NXT Standalone soll die Position halten

Sensor:

- NXT Gyro Sensor –  
Detektion des Neigungswinkels
- Impulsgeber des Servo Motors –  
Geschwindigkeitsdetektion / Radstand

Aktor:

- NXT Servo Motor – Korrektur der Neigung



# Anwendungsanalyse

---

Ziel:

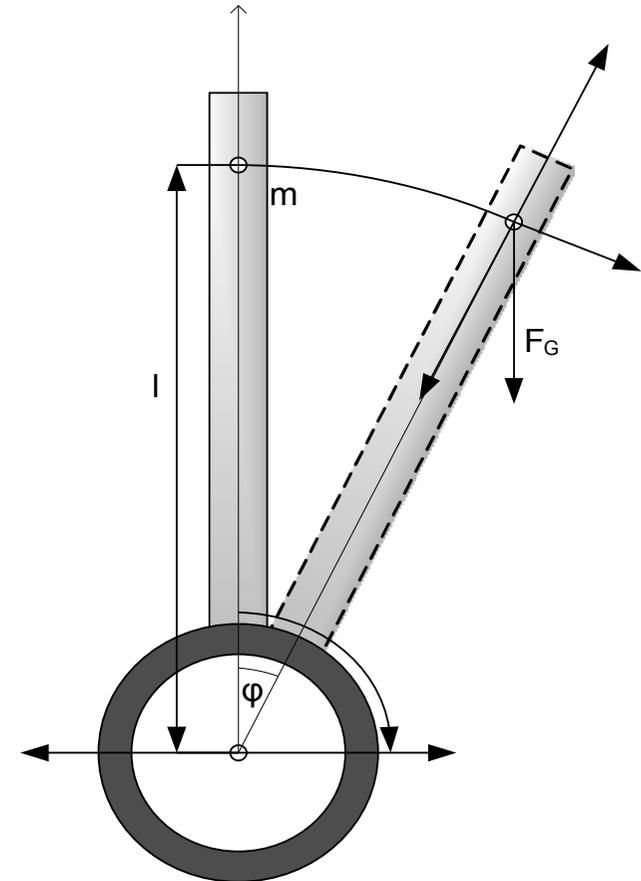
- NXT Standalone soll nicht umkippen
- NXT Standalone soll Hindernis ausweichen

Sensor:

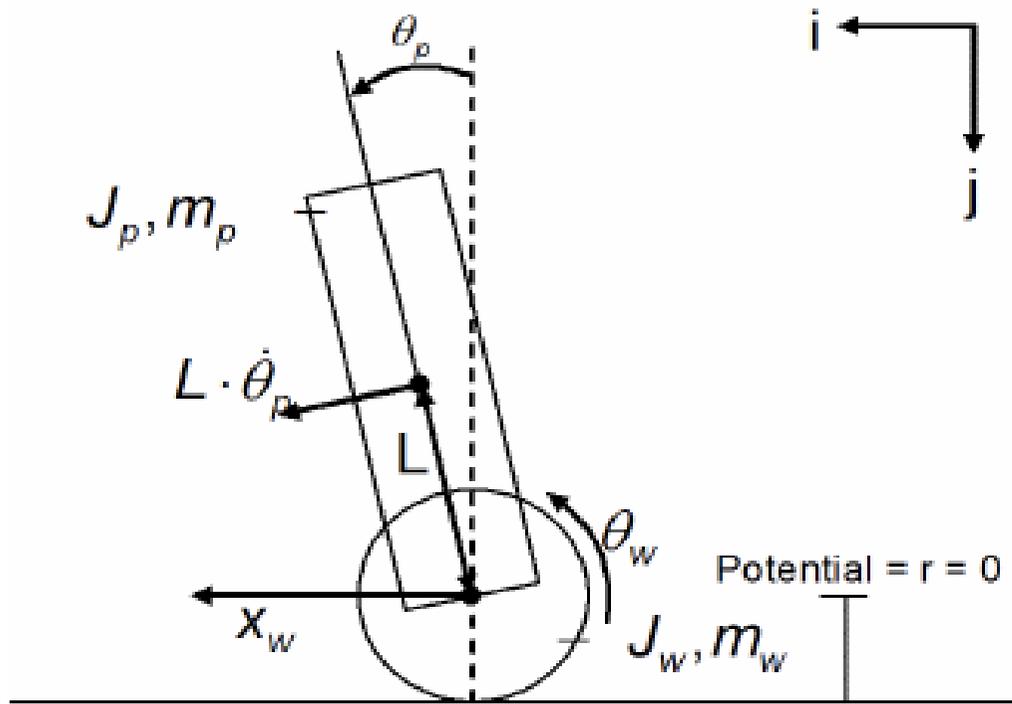
- NXT Gyro Sensor –  
Detektion des Neigungswinkels
- Impulsgeber des Servo Motors –  
Geschwindigkeitsdetektion / Radstand
- Ultraschall Sensor und Bluetooth

Aktor:

- NXT Servo Motor – Korrektur der Neigung



# Physikalisches Modell Lagrange Mechanik



Lagrange Mechanik

$$M_p = J'_p \cdot \ddot{\theta}_p + m_p \cdot r \cdot L \cdot \ddot{\theta}_w - m_p \cdot g \cdot L \cdot \theta_p$$

$$M_w = J'_w \cdot \ddot{\theta}_w + m_p \cdot r \cdot L \cdot \ddot{\theta}_p$$

# Physikalisches Modell Übertragungsfunktion

---

Radmoment muss Pendelmoment entgegen wirken

$$M_w = M$$

$$M_p = -M$$

Einführung der Reibung in den Antriebskomponenten

$$M_w = M - \mu \cdot \omega$$

$$M_p = -M + \mu \cdot \omega$$

Einführung der Rollreibung der Räder

$$M_w = M - \mu \cdot \omega - \delta \cdot \dot{\theta}_w$$

# Physikalisches Modell Übertragungsfunktion

---

Laplace Transformation

$$-M = (m_p \cdot r \cdot L \cdot s^2 - \mu \cdot s) \cdot \theta_w + (J'_p \cdot s^2 + \mu \cdot s - m_p \cdot g \cdot L) \cdot \theta_p$$

$$M = (J'_w \cdot s^2 + \mu \cdot s + \delta \cdot s) \cdot \theta_w + (m_p \cdot r \cdot L \cdot s^2 - \mu \cdot s) \cdot \theta_p$$

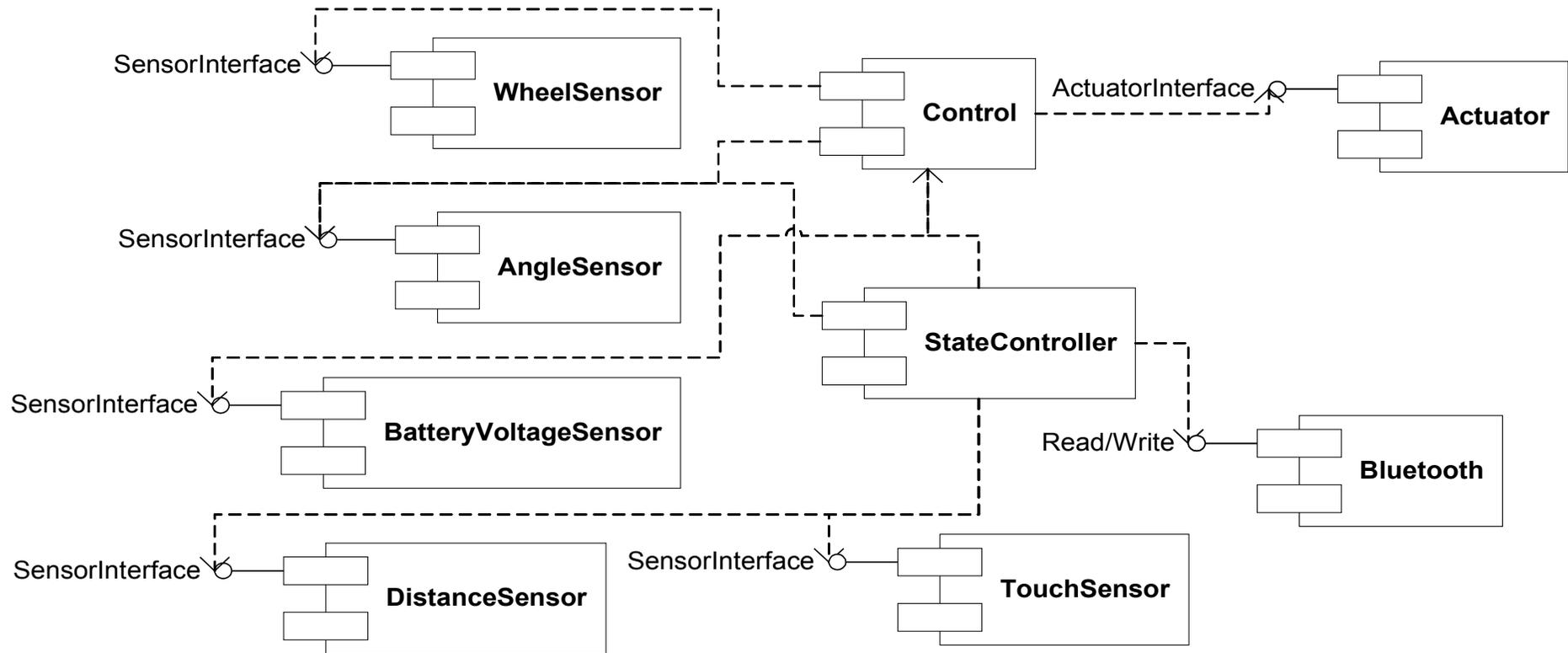
Übertragungsfunktionen

$$\frac{\theta_p(s)}{M(s)} \quad \frac{\theta_w(s)}{M(s)} \quad \frac{\omega(s)}{M(s)} \quad \omega = s \cdot \theta_w - s \cdot \theta_p$$

=> Nutzung der Regelungstechnik im Zustandsraum

# Komponenten und Schnittstellen

## Komponentenübersicht und Abhängigkeiten



# Komponenten und Schnittstellen

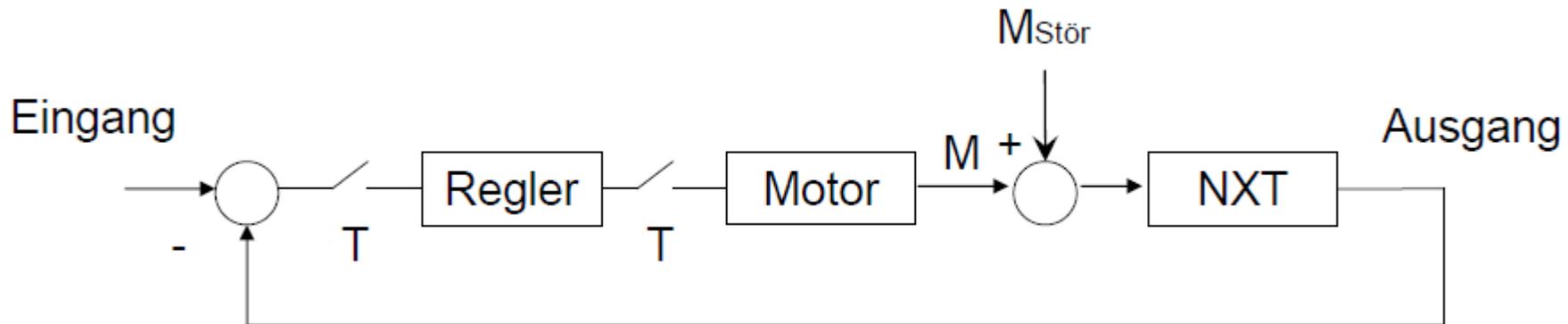
---

Einführung einer einfachen Schnittstelle für die Sensoren:

- S16 GetValue( )
  - Liefert den letzten gesampelten Wert zurück
- Sample( )
  - Liest einen neuen Wert vom Sensor aus

# Bestimmung der Regelstrecke

---



Probleme:

- 2 Abtastglieder, eine Störgröße und eine Rückführung in der Regelstrecke, somit sehr aufwendige und fehleranfällige Berechnung
- Drehzahl-Drehmoment-Kennlinie des Motors ist unbekannt
- Ein und Ausgänge sind mehrdimensional somit Berechnung mit Matrizen

# Bestimmung der Regelungsstrecke Fazit

---

- Probleme beim Aufstellen der Regelungsstrecke
  - Festlegung der zwingend nötigen physikalischen Eigenschaften ist schwer lösbar
  - Kleine Fehler können das Verhalten der Regelstrecke signifikant beeinflussen
  - Für eine Praktikum der Echtzeitsystem sehr Regelungstechnik lastig
  - Hier sollte eine feste Regelungsstrecke vorgegeben werden um die Komplexität zu vermindern

=> Benutzen eines Vorgegebenen Reglers

# Mathworks NXTway-GS Modell

---

- Abbildung des kompletten NXT in Matlab
  - Physikalisches Modell erfasst
    - Reibungen
    - Unterschiede der NXT Bauteile
  - Regler
  - State Machine
  - OSEK
  
- Sehr komplexes System Simulation nicht möglich

# Mathworks NXTway-GS Modell

---

- Funktionen des Modells
  - Stehen
  - Fahren
    - Synchronisation der beiden Räder
    - Kurven
  - Erkennung von Hindernisse
  - Kartographierung
  - Fernsteuerung
  - Simulation der NXT Bewegungen in 3D

# Mathworks NXTway-GS Regler Balancer.c

---

- Generierung des Reglercodes aus Matlab
- Periodendauer des Reglers nach Generierung fest (4ms)
  - Feste Periodendauer wird durch digitale Integration und Differentiation erzwungen
  - Beispiel: Differentiation

# Komponententest und WCET-Bestimmung

---

## Übersicht Testkonzept

- Testkonzept
- informelle Tests
  - Inspection basiert
  - LCD Ausgaben
  - Feststellung der Motorlaufrichtung

## Automatisierung

- Verknüpfung von Komponenten ( Wheel / Aktuator )
- Anforderung:
  - Korrekte Funktion von Messwertaufnahme / Stellwertsetzen

# Komponententest und WCET-Bestimmung

---

## AngleSensor

- Sensor liefert Winkel und Winkelgeschwindigkeit zurück
  - Kippen nach vorn → Winkel-/ geschwindigkeit nimmt positiv zu
  - Kippen nach hinten → Winkel-/ geschwindigkeit nimmt negativ zu
- Verifikation durch Überprüfung des am LCD ausgegebenen Wertes

## WheelSensor

- Sensor liefert den zurückgelegten Weg zurück (in Grad)
  - Bewegung der Räder → liefert zurückgelegten Weg
  - Bewegung in andere Richtung → Abnahme des Werts
- Verifikation durch Überprüfung des am LCD ausgegebenen Wertes

# Komponententest und WCET-Bestimmung

---

## BatteryVoltageSensor

- Sensor liefert den aktuellen Wert der Batteriespannung
  - Spannung im Bereich 7-9V
- Verifikation durch Überprüfung des am LCD ausgegebenen Wertes.

## ButtonSensor

- Sensor liefert den Zustand des Tasters zurück
  - Taster wurde gedrückt /nicht gedrückt

# Komponententest und WCET-Bestimmung

---

## StateController

- Zustandsumschaltung
  - Stehen, Fahren, Ausweichen, Notaus
  - Kommandoverarbeitung zur Zustandsumschaltung (Bluetooth)
  - NXT gerät in eine nicht kompensierbare Schiefelage → Notaus
- Verifikation des ausgegebenen Zustands

## Actuator

- Aktorkomponente steuert die beiden Motoren an
  - Setzen der Motorspannung (PWM-Wert)
  - Auslesen des aktuellen Wertes und Ausgabe am LCD
- Verifikation durch Überprüfung des am LCD ausgegebenen Wertes

# Komponententest und WCET-Bestimmung

---

## balance\_control

- Verifikation der Komponente im System
- Verifikation durch manuelles Bewegen des NXT

## DistanceSensor

- Ausgabe der Abstandsanzeige auf dem LCD
- Verifikation durch Überprüfung des am LCD ausgegebenen Wertes

## Bluetooth

- Ausgabe der empfangenen Daten auf dem LCD
- Verifikation durch Überprüfung des am LCD ausgegebenen Wertes

# WCET-Bestimmung

---

1 cycle = 1/48MHz = 21ns

Wheelsensor.Sample	41 cycles / 0.85µs
AngleSensor.Sample	957 cycles / 19.94µs
Actuator.Set	252 cycles / 5.25µs
VoltageSensor.GetValue	7 cycles / 0.15µs
balance_control	8499 cycles / 177.06µs
Wheelsensor.GetValue	8 cycles / 0.17µs
AngleSensor.GetValue	10 cycles / 0.21µs
AngleSensor.GetAngle	8 cycles / 0.17µs
StateController.Action	47 cycles / 0.98µs
StateController.Update	7385 cycles / 153.85µs
StateController.CheckBT	7299 cycles / 152.06µs

# Komposition

---

- InitTask
  - Einmalige Ausführung beim Start
  - Initialisiert die Komponenten
- SensorTask (4 ms Periode)
  - Regler ist auf 4 ms Periode diskretisiert
  - Sampling Rate vom GyroSensor 300Hz
- ControlTask
  - Aktivierung durch SensorTask (Synchronisation mit SensorTask)
- ControllerTask (50 ms Periode)
  - Nicht wahrnehmbare Verzögerung für Benutzer
    - Bluetooth
    - Emergency off
  - Sonarsensor max. Samplingfrequenz: ~40ms

# Komposition

---

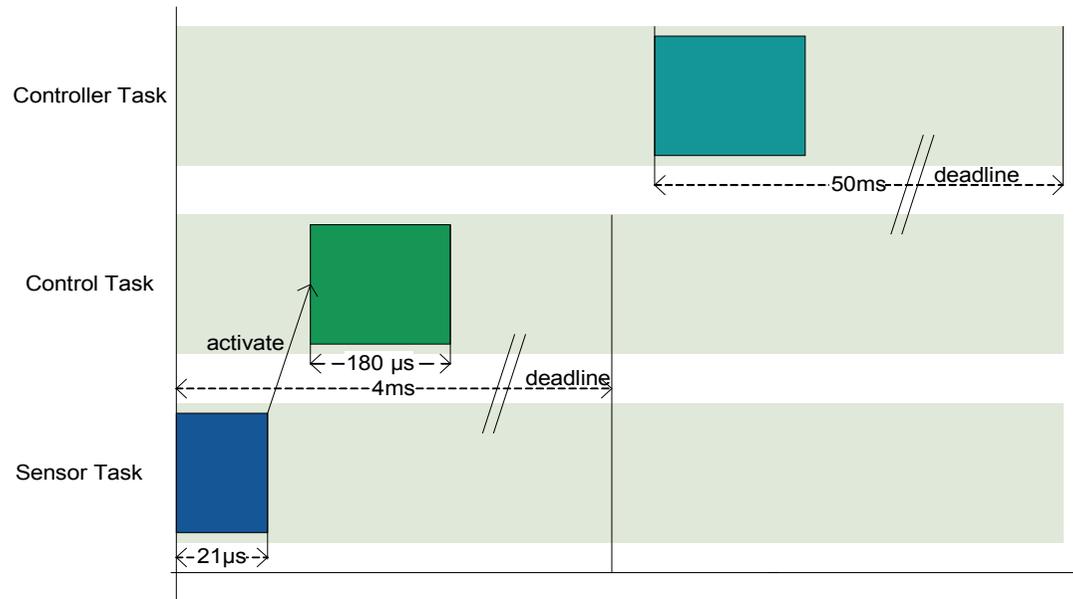
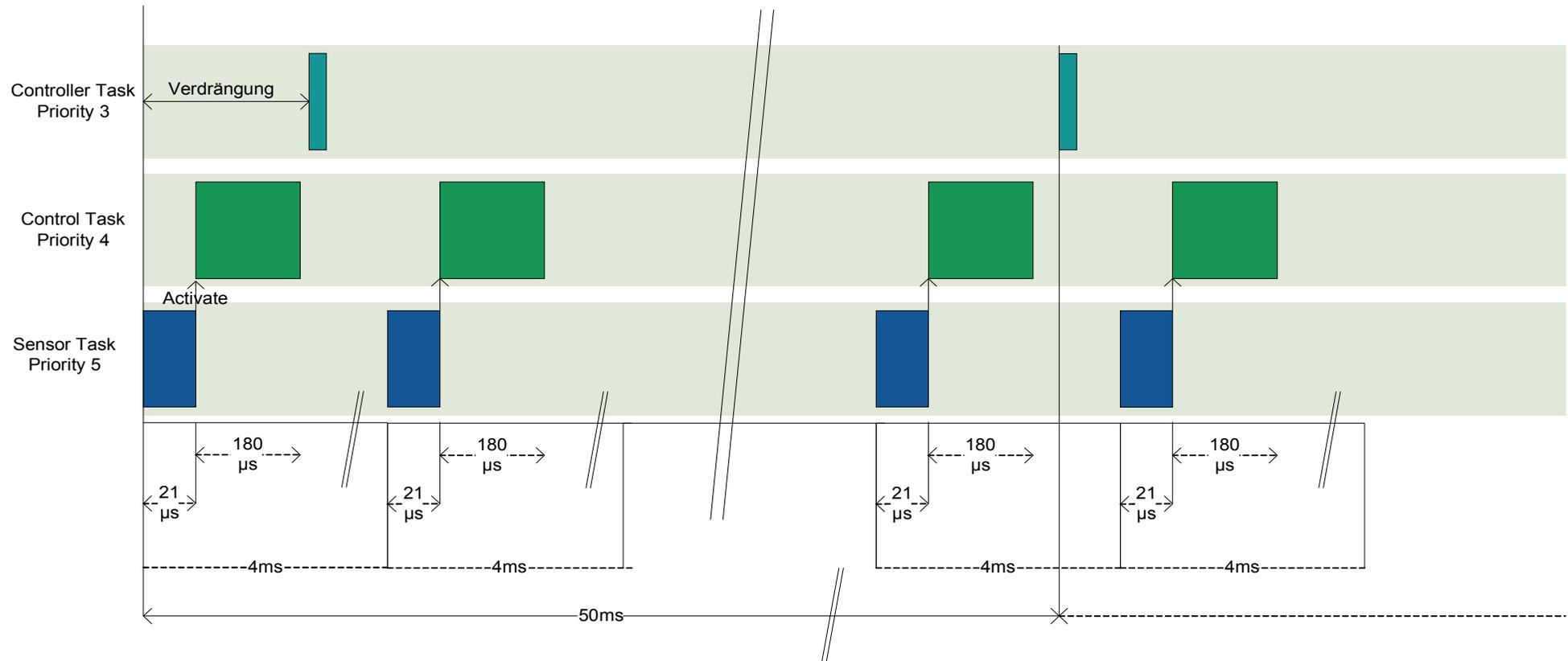


Abbildung der Komponenten:

- SensorTask: AngleSensor, WheelSensor, ButtonSensor
- ControlTask: balance\_control, Actuator
- ControllerTask: StateController.Update/Action, DistanceSensor

# Komposition



## Zeitliche Eigenschaften und Verdrängung

- SensorTask aktiviert ControlTask → Synchronisierung
- ControllerTask ist nicht synchronisiert → Verdrängung

# Akzeptanztest

---

- NXT soll für den Kunden bedienbar sein
  - NXT reagiert jederzeit auf Eingaben des Bedieners
  - NXT reagiert auf Ein/Ausschaltknopf
  - Eingaben können per Bluetooth erfolgen
- Demonstration im Anschluss

# Kleiner Test

---

