

Demonstration der Funktionsweise eines Segways mit Hilfe der Programmierung eines Lego Mindstorms Roboters.

Präsentationsprüfung
von Peter Segieth
In den Fächern Informatik und Physik



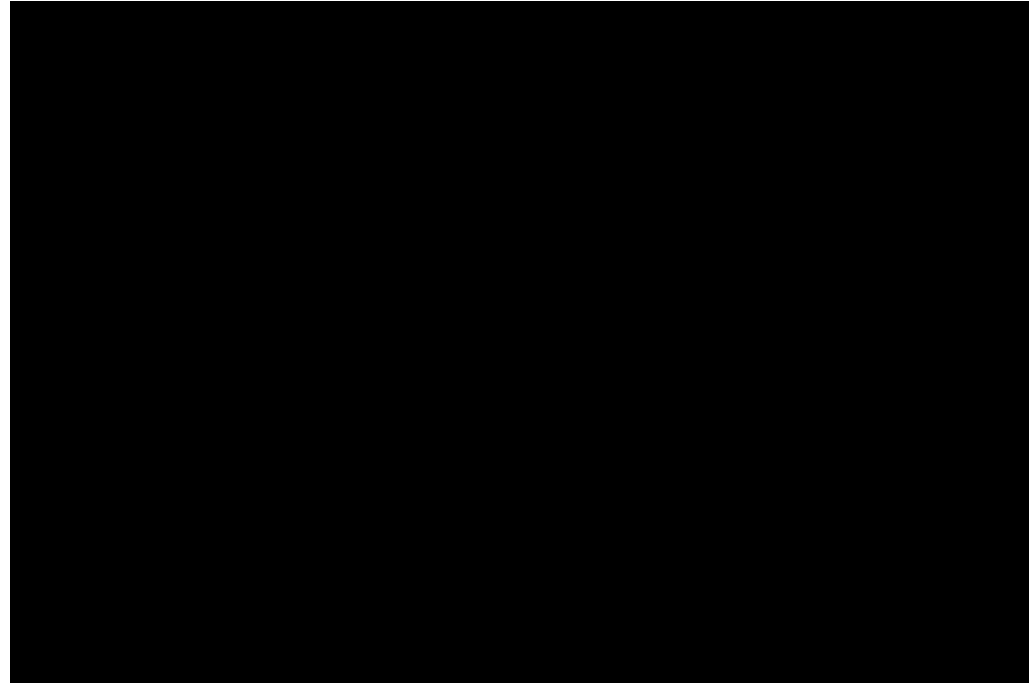
Gliederung



1. Was ein Segway ist
 1. Video
 2. Wie man ihn steuert
 3. Problemstellung
2. Die Physik hinter einem Segway
 1. Das (inverse) Pendel
 2. Der Gleichgewichtssinn
3. Informationstechnische Lösung des Problems
 1. Der Regelkreis
 2. Vorstellung meiner Modellversion
 3. Das Programm als Struktogramm
4. Unterschiede meiner Modellversion und des echten Segways
5. Fazit



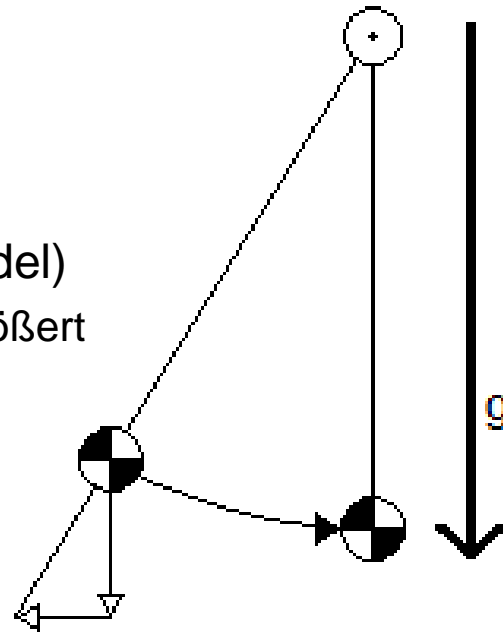
1. Der Segway Personal Transporter



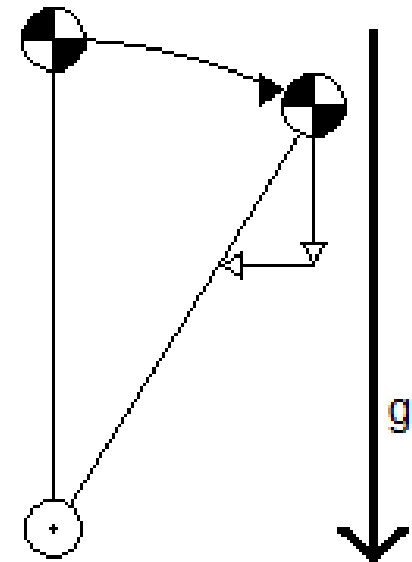
- „Einen Segway zu fahren ist so intuitiv, dass es sich anfühlt, als ob das Ding irgendwie an das Zentralnervensystem angeschlossen wäre.“ – Fortune Magazine
- Wie kann sich der Segway der Erdanziehungskraft widersetzen?

2. Die Physik - Das Pendel

- Stabile Lage
 - Das Rückstellmoment verkleinert die Auslenkung
- Instabile Lage (inverses Pendel)
 - Das Rückstellmoment vergrößert die Auslenkung.



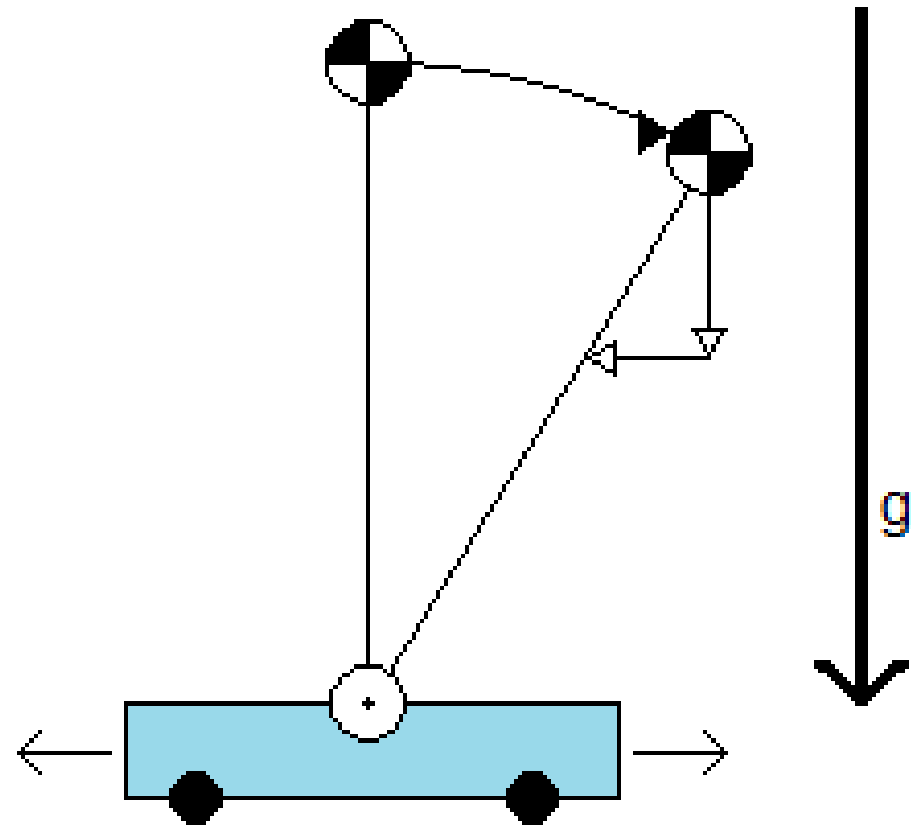
-> Ein inverses Pendel fällt im Regelfall um!



Das inverse Pendel – Die Lösung

- Aktives Stabilisieren
 - Drehpunkt unter den Schwerpunkt bewegen
 - Rückstellmoment zu Null gemacht

-> Das inverse Pendel steht aufrecht!



3. Die IT-Lösung - Der Gleichgewichtssinn

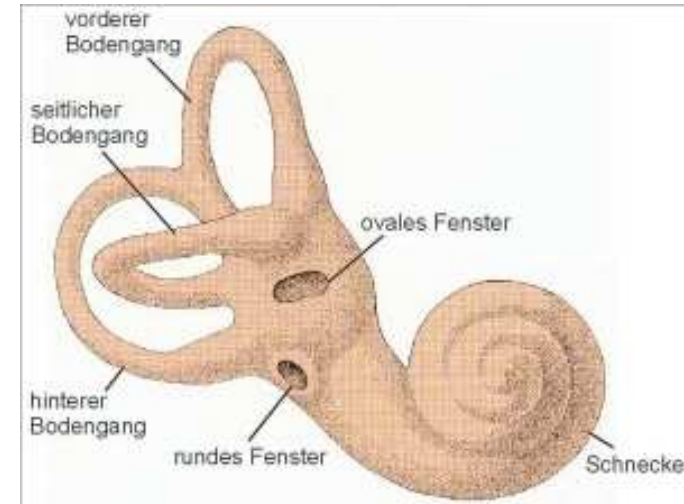
- Beim Menschen angeboren im Ohr
- Im Segway: Gyroskopsensoren

Vorteile	Nachteile
Genauere Winkelbestimmung	Drift
Resistent gegen Beschleunigungen	Offset

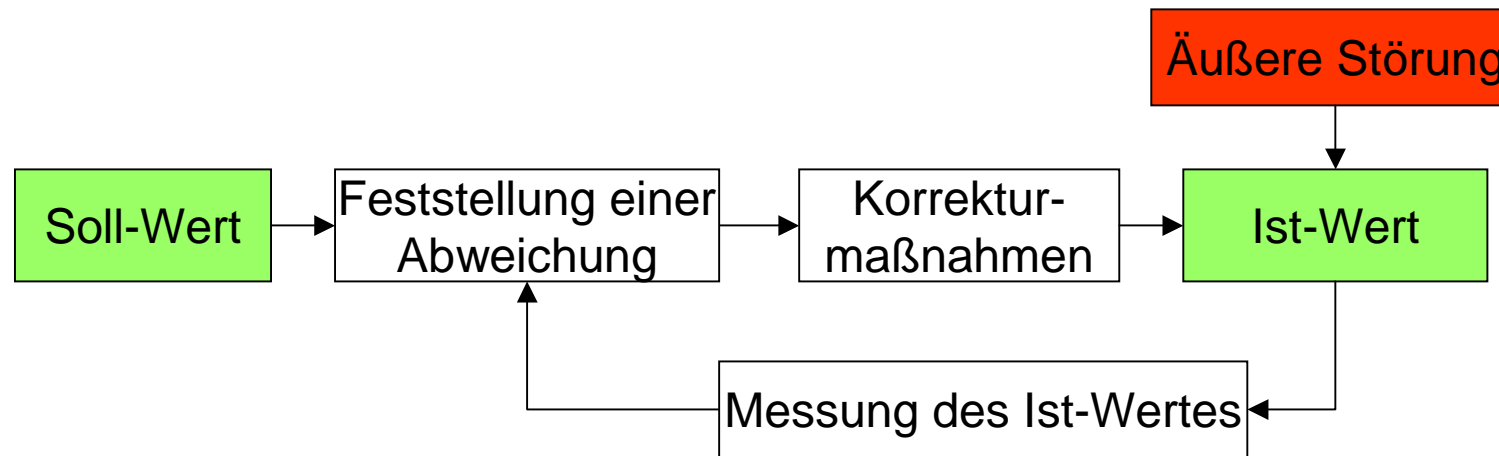
– Geben Winkelgeschwindigkeit zurück

$$\alpha = \int_0^t \omega dt = \int_0^t \frac{\alpha}{t} dt$$

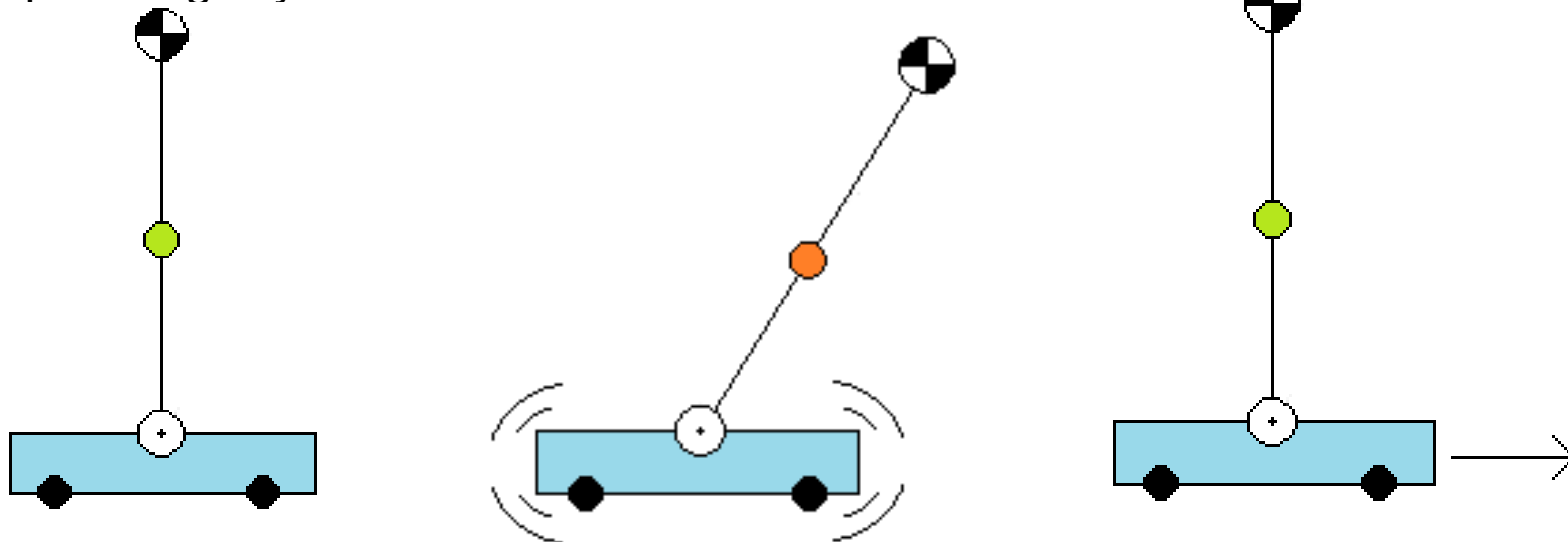
– Oft auslesen und Werte zusammenaddieren



Der Regelkreis



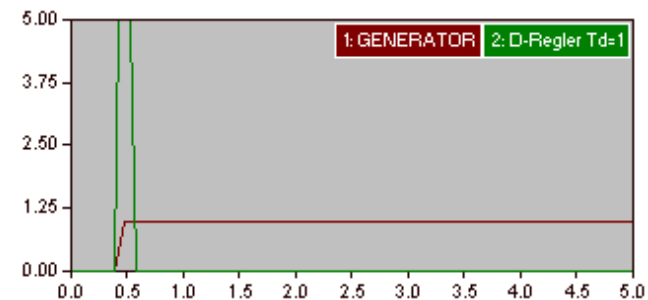
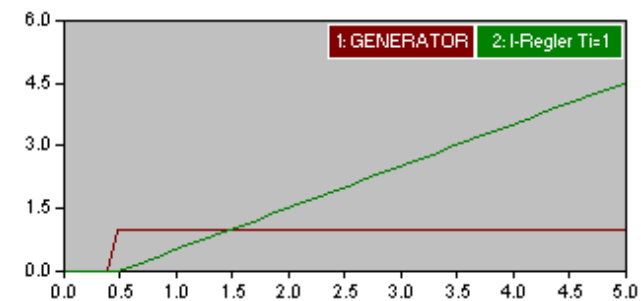
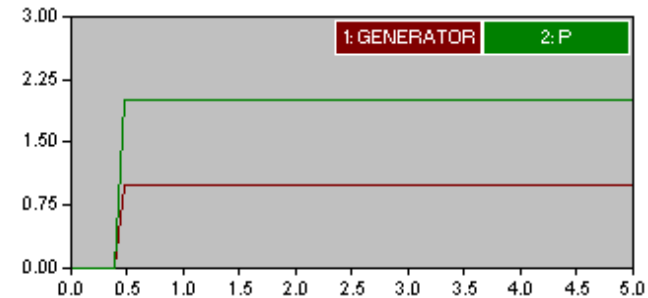
Beispiel: Segway



PID-Regler

- Proportionalwirkender Anteil
 - Mittelschnell
 - Bleibende Regelabweichung
- Integralwirkender Anteil
 - Sehr langsam
- Differentialwirkender Anteil
 - Sehr schnell
 - Kann zu hohe Werte ausgeben
 - Vorbeugung eines Fehlers

$$y(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot e'(t)$$

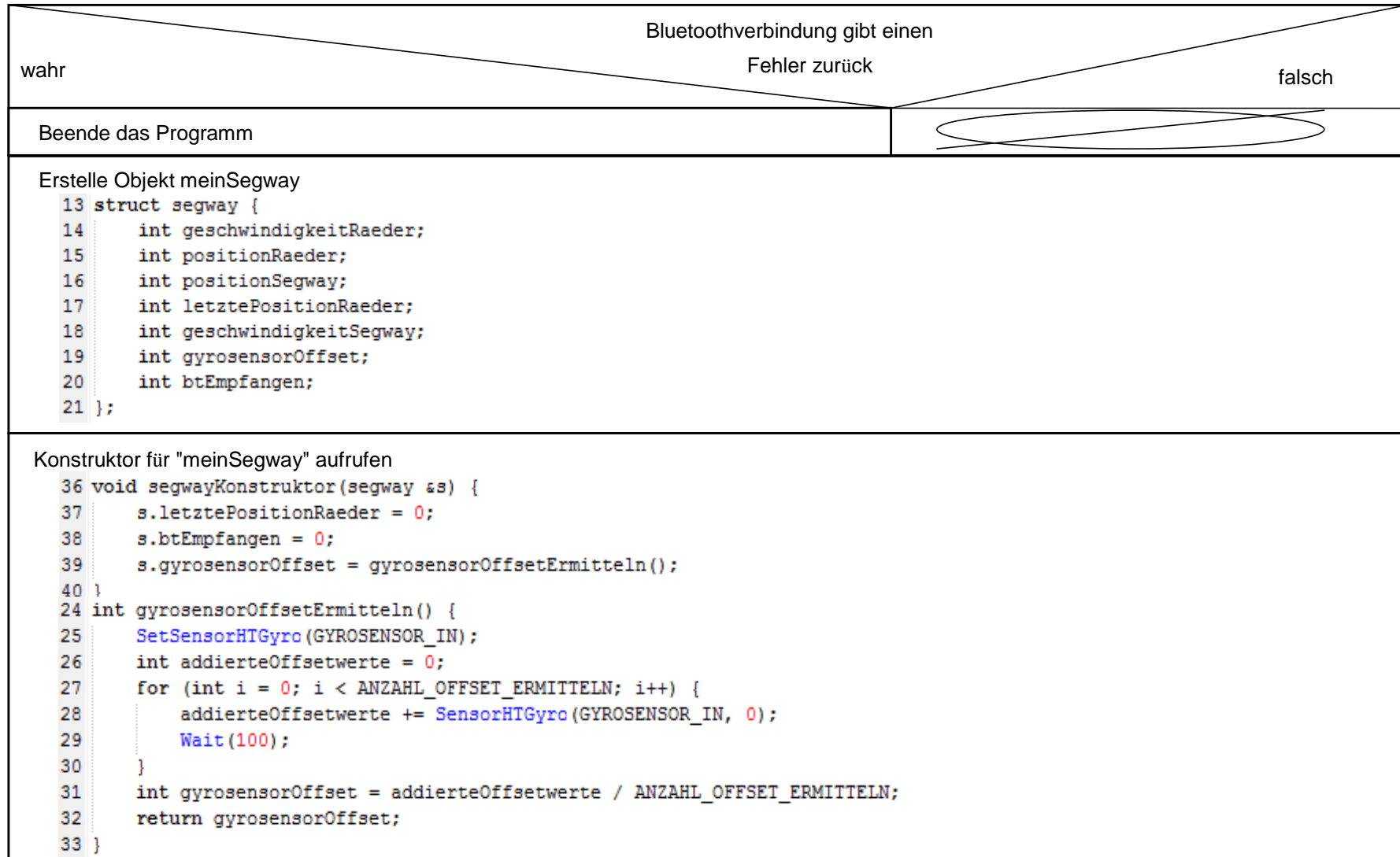


Mein Lego-Modell

- Inverses Pendel
- Gyroskopsensor als Gleichgewichtsorgan
- PD-Softwareregler
 - Ist sehr schnell
 - Kein I-Anteil, weil dieser ist sehr langsam
- Fernsteuerung per Bluetooth
- 2 Getriebemotoren und Rotationszähler



Struktogramm des Programms I.



Struktogramm des Programms II.

so lange die Bluetoothnachricht nicht gleich 9 ist

letzte Bluetoothnachricht aus dem Nachrichtenbuffer auslesen

Berechnung der Motorenleistung für die Aufrechthaltung oder Vor-/ Rückwärtsbewegung

```
108 int motorLeistungErmitteln(segway *s) {
109     int motorLeistung;
110     const int gyrosensorOffset = s.gyrosensorOffset;
111
112     // Neigungs- / Winkelgeschwindigkeit (Differenzialer Teil des Segways)
113     s.geschwindigkeitSegway = SensorHTGyro(GYROSENSOR_IN, gyrosensorOffset);
114
115     // Neigungswinkel (Proportionaler Teil)
116     s.positionSegway += s.geschwindigkeitSegway + zusatzVor(s.btEmpfangen);
117
118     // Gesamter Drehwinkel der Räder (Proportionaler Teil der Räder)
119     s.positionRaeder = (MotorTachoCount(LINKER_MOTOR_OUT) + MotorTachoCount(RECHTER_MOTOR_OUT)) / 2;
120
121     // Winkelgeschwindigkeit der Räder (Differenzialer Teil)
122     s.geschwindigkeitRaeder = s.positionRaeder - s.letztePositionRaeder;
123     s.letztePositionRaeder = s.positionRaeder;
124
125     // Berechnung der Motorenleistung durch einen PD-Regler
126     motorLeistung = s.positionSegway * PS_FAKTOR + s.geschwindigkeitSegway * GS_FAKTOR
127         + s.positionRaeder * PR_FAKTOR + s.geschwindigkeitRaeder * GR_FAKTOR;
128     motorLeistung += sign(motorLeistung) * ML_VERSTAERKUNG;
129     return motorLeistung;
130 }
```

Motoren ansteuern mit Leistungszusätzen für evtl. Richtungsänderung des Segways

Wartezeit um den PD-Regler zu dämpfen

Stoppe die beiden Motoren

4. Unterschiede des Modells und des Segway PT

Segway PT	Lego-Segway
Zur Sicherheit ist alles redundant vorhanden	
5 Gyroskopsensoren 2 Beschleunigungssensoren	1 Gyroskopsensor
Direkt gesteuert	Ferngesteuert
Kálman-Filter gegen Messungenauigkeiten	

Kálmán-Filter

- Nutzen: Messfehler beheben
- Koppelung zweier Sensoren
- Schätzt die genauen Werte
- Beispiel: Positionsbestimmung von Flugzeugen



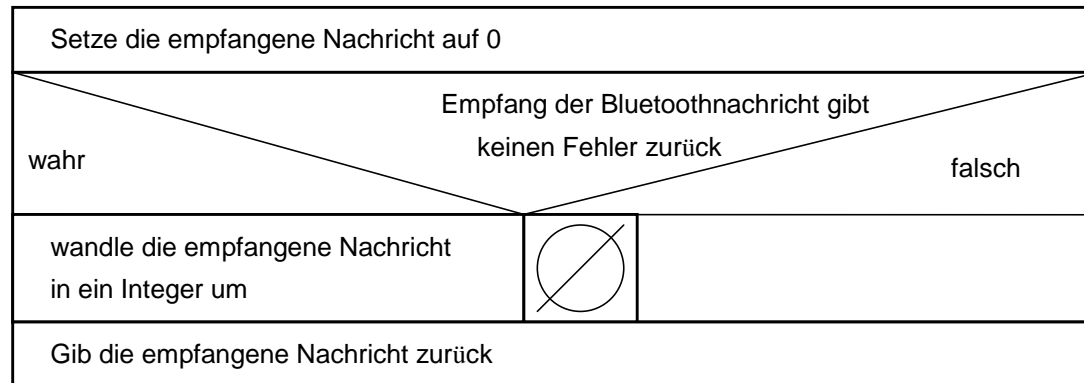
5. Fazit

Ein Segway benötigt eine komplizierte Regelung um stabil zu fahren. In der Vorbereitung auf die Präsentationsprüfung habe ich die physikalischen Hintergründe analysiert. Mit dem Legomodell konnte die Funktionsweise erfolgreich demonstriert werden.

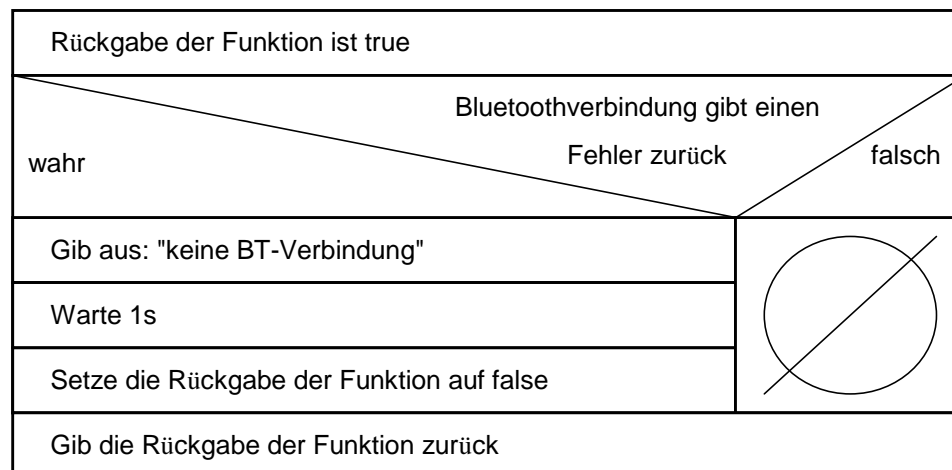


Weitere Struktogramme I.

int Bluetoothmailbox auslesen



boolean BT Verbindungstest





Vielen Dank für Ihre Aufmerksamkeit!

Weitere Struktogramme II.

int Leistungszusätze für den linken Motor

empfangene Nachricht		
4	6	sonst
gib -10 zurück	gib 10 zurück	gib 0 zurück

int Leistungszusätze für den rechten Motor

empfangene Nachricht		
4	6	sonst
gib 10 zurück	gib -10 zurück	gib 0 zurück

int Vor-/ Rückwärtszusätze

empfangene Nachricht		
8	2	sonst
gib -3 zurück	gib 3 zurück	gib 0 zurück

Weitere Struktogramme III.

int Offset des Gyroskopsensors ermitteln

Bestimme an welchem Eingang der Sensor angeschlossen ist

durchlaufe 10 Mal

addiere den AktuellenWert des Gyroskopsensors
zu einer Summe dazu

Warte 100 ms

Die Summe der Gyroskopsensorwerte durch die Anzahl der
Durchläufe teilen und zurückgeben

Segwaystruktur

Es muss vorhanden sein:

- aktuelle Winkelgeschwindigkeit der Räder
- aktuelle Position der Räder/ Gesamtdrehwinkel
- Position der Räder im letzten Scheifendurchlauf
- aktuelle Kippgeschwindigkeit des Segways
- aktueller Neigungswinkel des Segways
- letzte Empfangene Nachricht via BT

int Motorenleitung ermitteln

lies die Kippgeschwindigkeit des Segways aus

berechne den Neigungswinkel durch Addieren
der Kippgeschwindigkeiten, dazu den Zusatz
für evtl. Vor/ Rückwärtsbewegung

Ermittle die Position der Räder (gesamten Drehwinkel)
Dazu den Mittelwert beider Drehwinkel nehmen

Winkelgeschwindigkeit der Räder ausrechnen durch
Subtraktion der letzten Radposition

Berechnung der Motorenleistung durch einen PD-Regler

Verstärke die Motorenleistung je nach Batterieleistung

Gib die Motorenleistung zurück

void Segway Konstruktor

Setze die letzte Position der Räder auf 0

Setze die letzte empfangene Bluetoothnachricht auf 0

Ermittle den Offset des Gyroskopsensors

