



# *Fragen?*

- Wie komme ich effizient zu einem Programm?
- Wie beschreibe/dokumentiere ich meine Idee?



# *Schritte beim Programmieren*

- Idee, was der Roboter tun soll?
- EVA-Prinzip
  - E: Eingabe → Welche Daten werden über die Sensoren abfragen?
  - V: Verarbeitung → Wie werden die Daten verarbeitet?
  - A: Ausgabe → Wie werden die Daten ausgegeben?
- Aus welchen Teilen besteht das Programm?
  - Sequenzen** (Anweisungen werden nacheinander bearbeitet)
  - Entscheidungen** (Auswahl von einer von mehreren Anweisungen)
  - Wiederholungen** (Anweisungen wiederholen)
- Erstellen eines Flussdiagramms
- Das Programm wird geschrieben (implementiert)
- Das Programm wird getestet



## *Die Idee und das EVA-Prinzip*

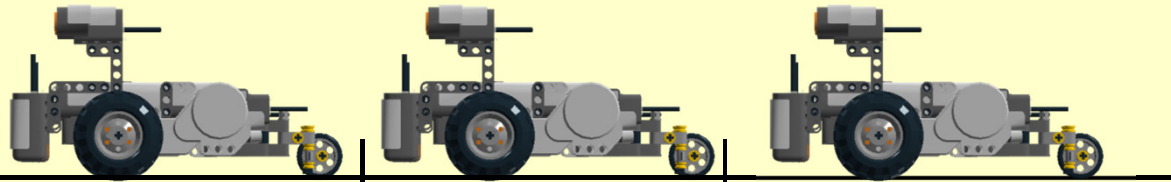
- Roboter fährt Person nach (Hündchen):



# *Analyse des Problems*



Fazit: Roboter kennt nach dem Einschalten seine Position nicht!



0

25

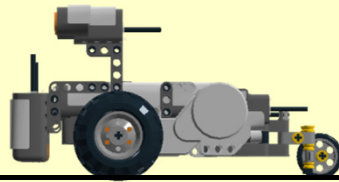
30



# *Lösung des Problems*



Roboter muss rückwärts fahren!



0

25

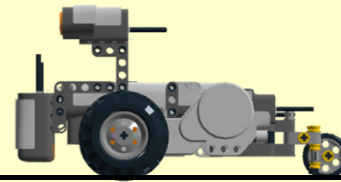
30



# *Lösung des Problems*



Roboter muss stoppen!



0

25

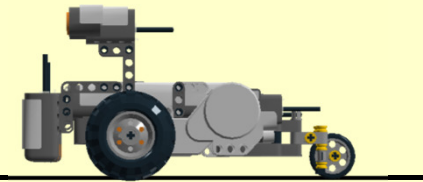
30



# *Lösung des Problems*



Roboter muss vorwärts fahren!



0

25

30



## *Die Idee und das EVA-Prinzip*

- Roboter fährt Person nach (Hündchen):
- Eingabe: Distanz über Ultraschallsensor
- Verarbeitung: Distanz  $< 25$  → rückwärts  
Distanz  $25 \leq \text{Distanz} \leq 30$  → stoppen  
Distanz  $> 30$  → vorwärts
- Ausgabe: Motoren rückwärts, stoppen oder vorwärts

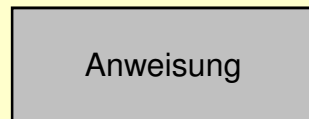




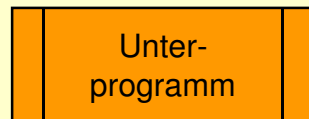
# *Elemente des Flussdiagramms*



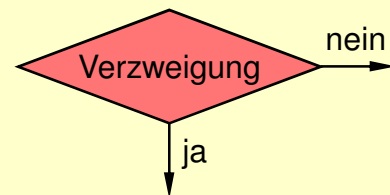
Oval: Start bzw. Stopp



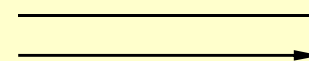
Rechteck: Anweisung



Rechteck mit doppelten, vertikalen Linien: Unterprogramm



Raute: Entscheidung/Wiederholung

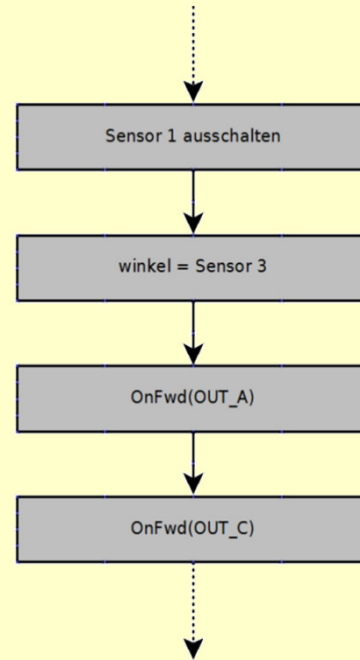


Linie bzw. Pfeil: Verbindung zum nächstfolgenden Element



# Beispiel Sequenz

Sequenz



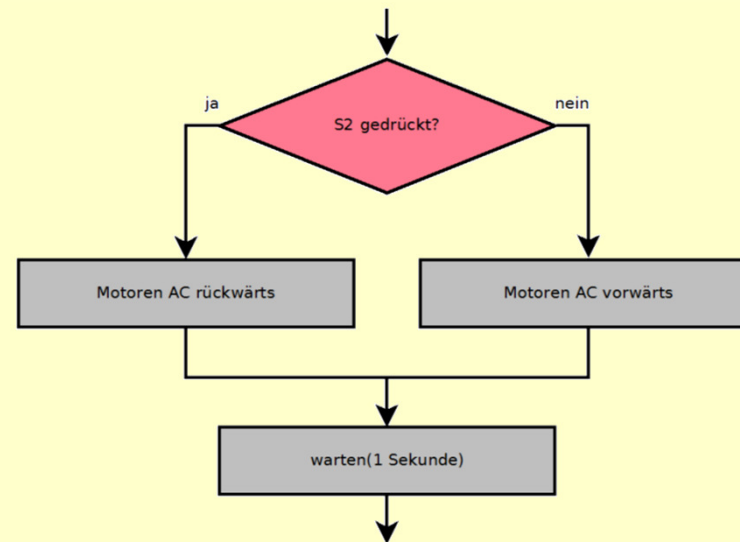
Umsetzung in NXT-G



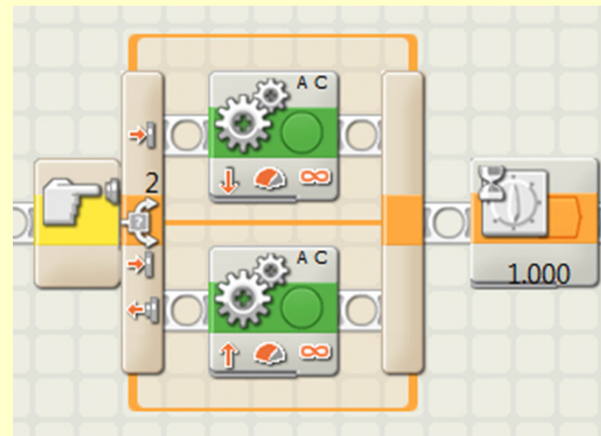


# Beispiel Entscheidung

Entscheidung  
(Schalter in NXT-G)



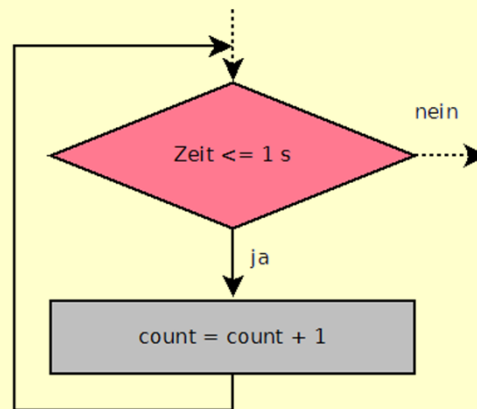
Umsetzung in NXT-G



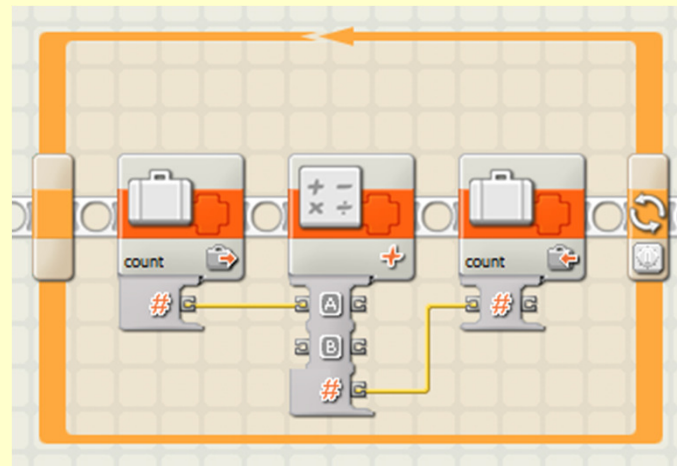


# Beispiel Wiederholung

Wiederholung



Umsetzung in NXT-G





## *Wieso ein Flussdiagramm?*

- Überblick gewinnen
- wichtige Elemente benennen
- Wechselwirkung darstellen
- enthält wichtigste Informationen
- Dokumentation der Software  
(auch Nichtprogrammierer können mitreden...)

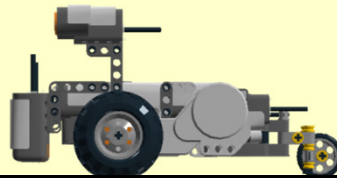


# *Auftrag: Flussdiagramm*



Sie erstellen ein Flussdiagramm:

Roboter fährt Person nach!



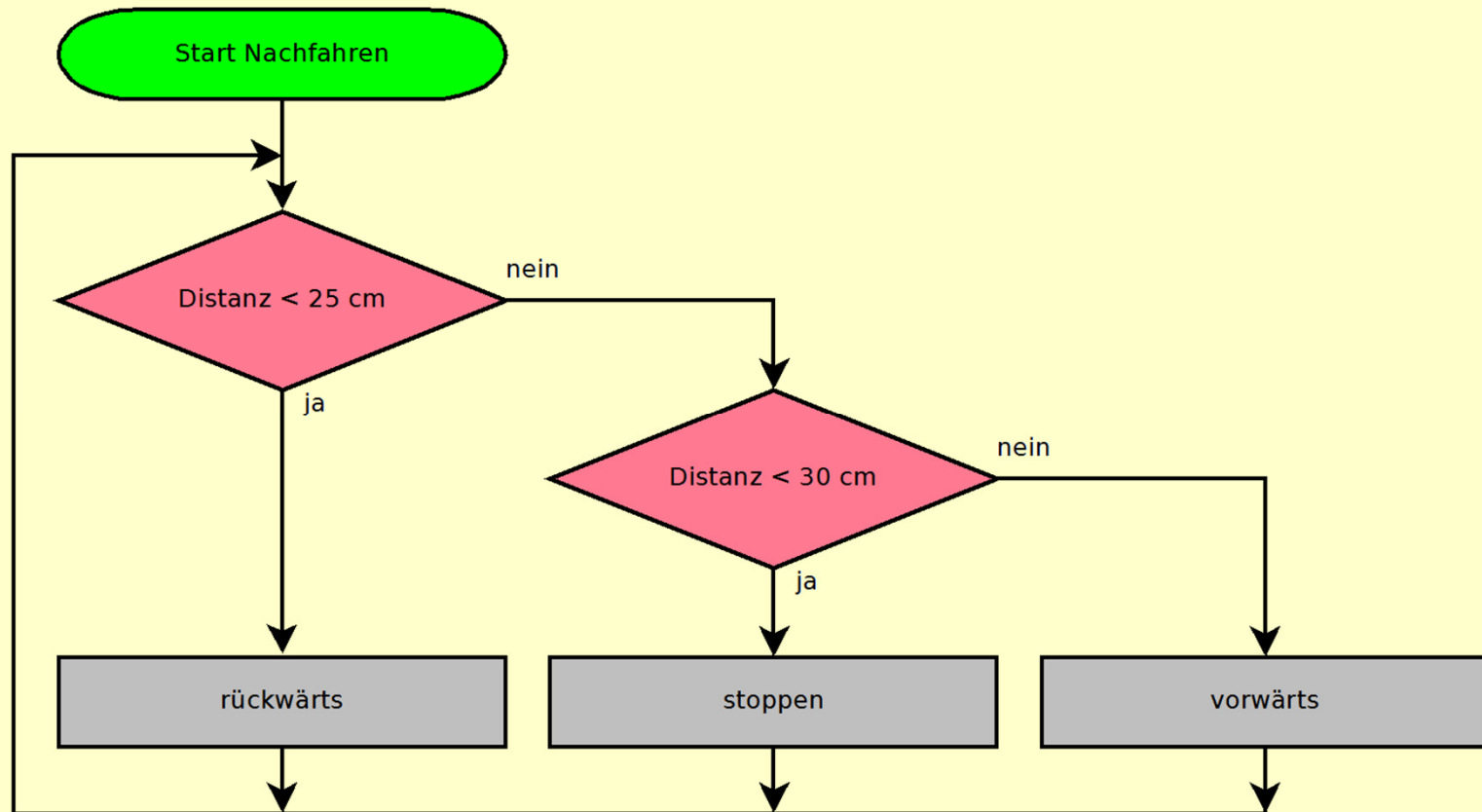
0

25

30

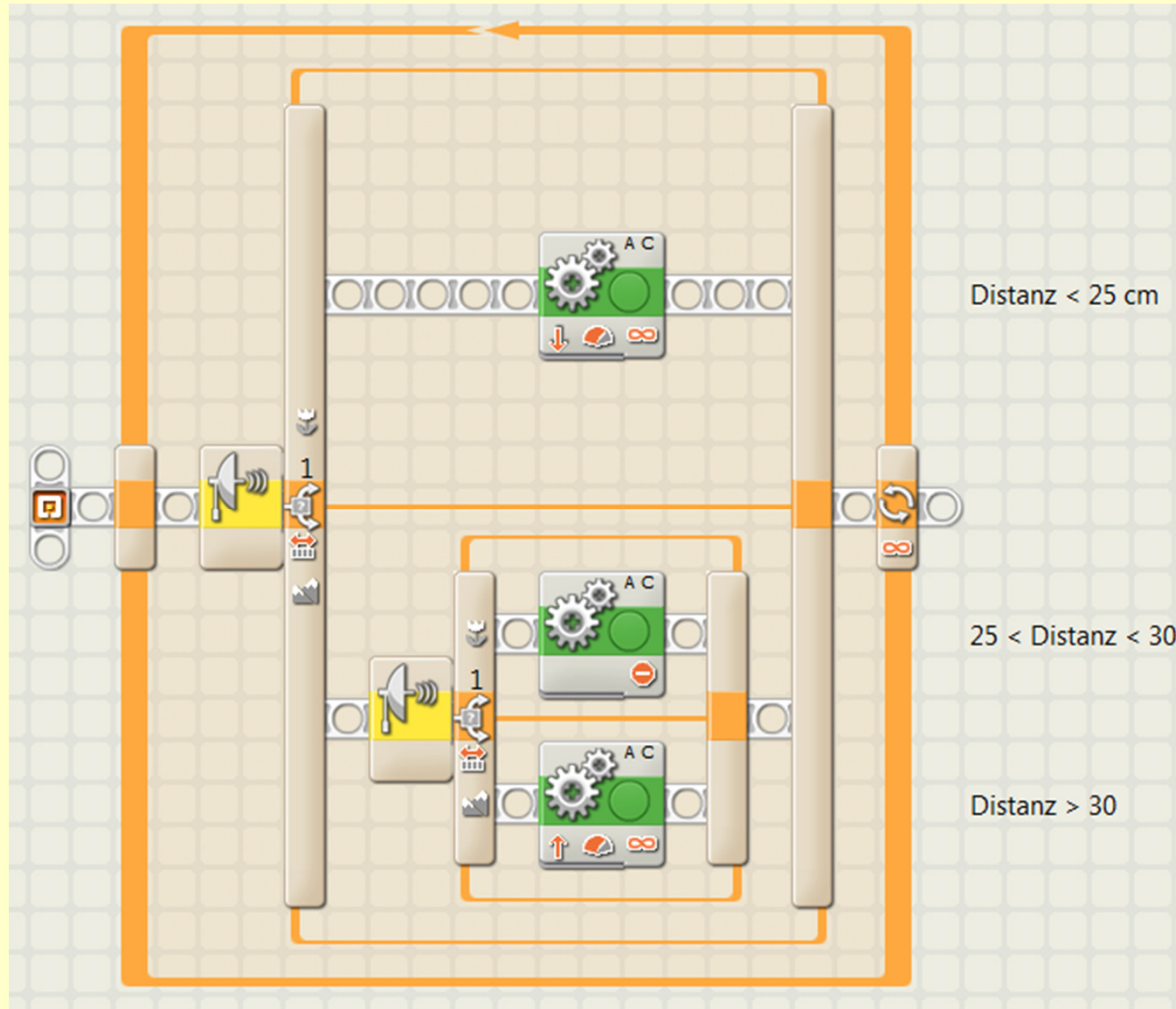


# *Eine mögliche Lösung:*





# Umsetzung in NXT-G





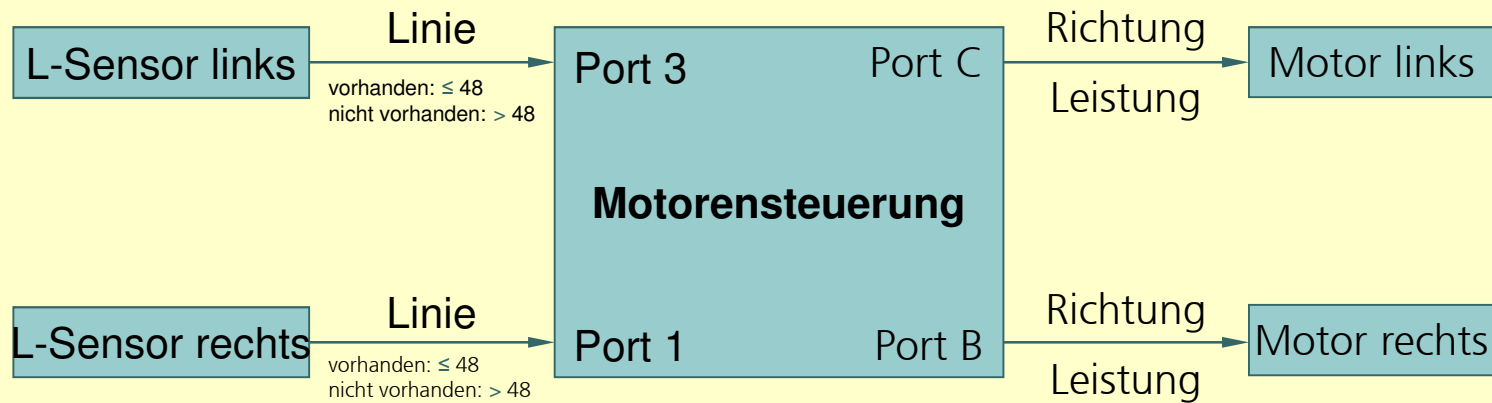


## *Das Problem?*

- Schwarzer Linie folgen mit zwei Lichtsensoren
  - beide Sensoren hell → geradeaus fahren
  - linker Sensor dunkel → nach links drehen
  - rechter Sensor dunkel → nach rechts drehen



# *Funktionale Betrachtung*



Wie entwickle ich ein Robolab- bzw. NXT-G-Programm?



## *Wieso ein Flussdiagramm?*

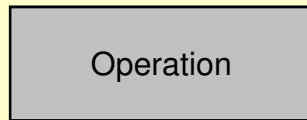
- Überblick gewinnen
- wichtige Elemente benennen
- Wechselwirkung darstellen
- enthält wichtigste Informationen
- Dokumentation der Software  
(auch Nichtprogrammierer können mitreden...)



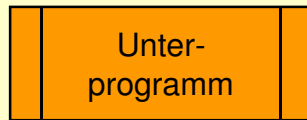
# *Elemente des Flussdiagramms*



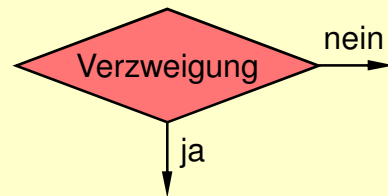
Oval: Start bzw. Stopp



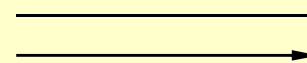
Rechteck: Operation



Rechteck mit doppelten, vertikalen Linien: Unterprogramm



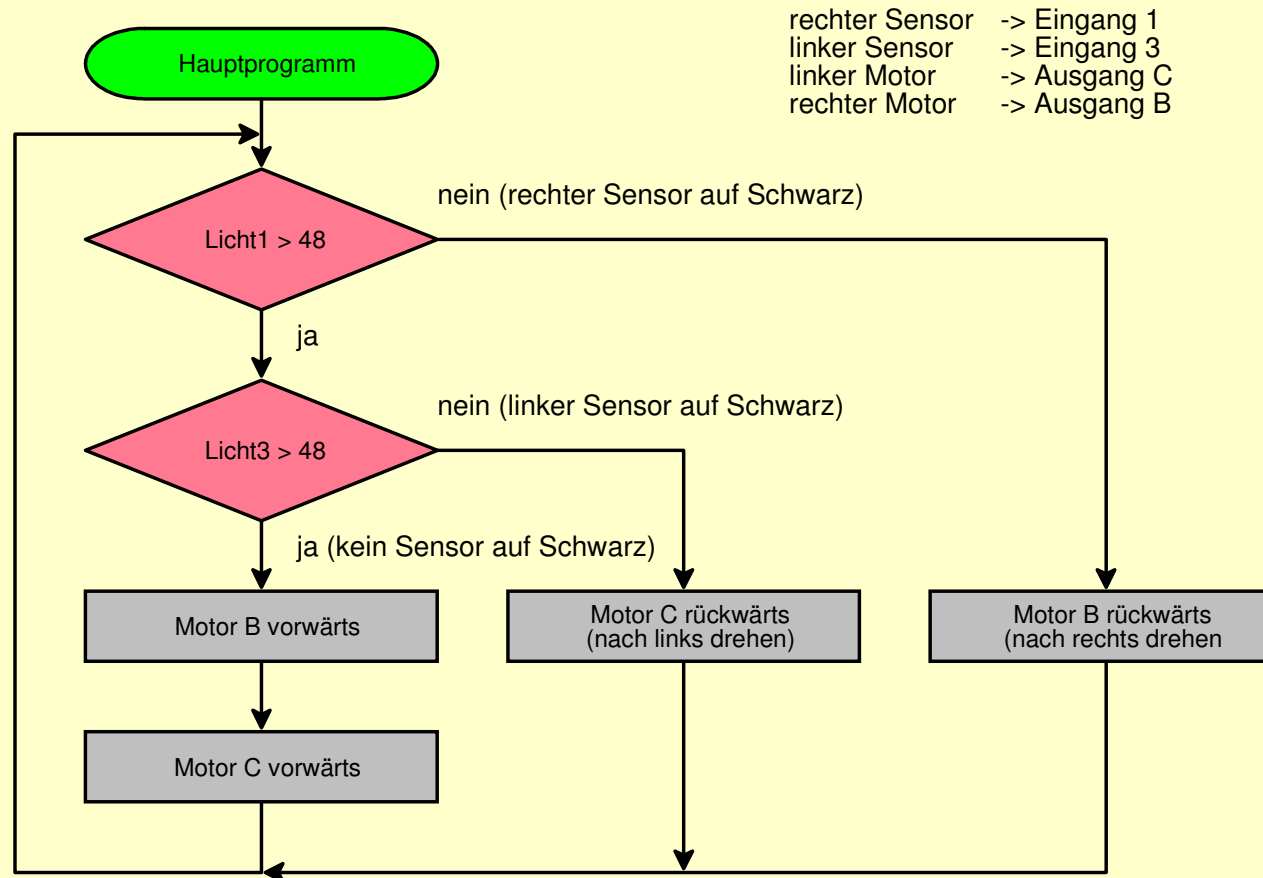
Raute: Verzweigung



Linie bzw. Pfeil: Verbindung zum nächstfolgenden Element

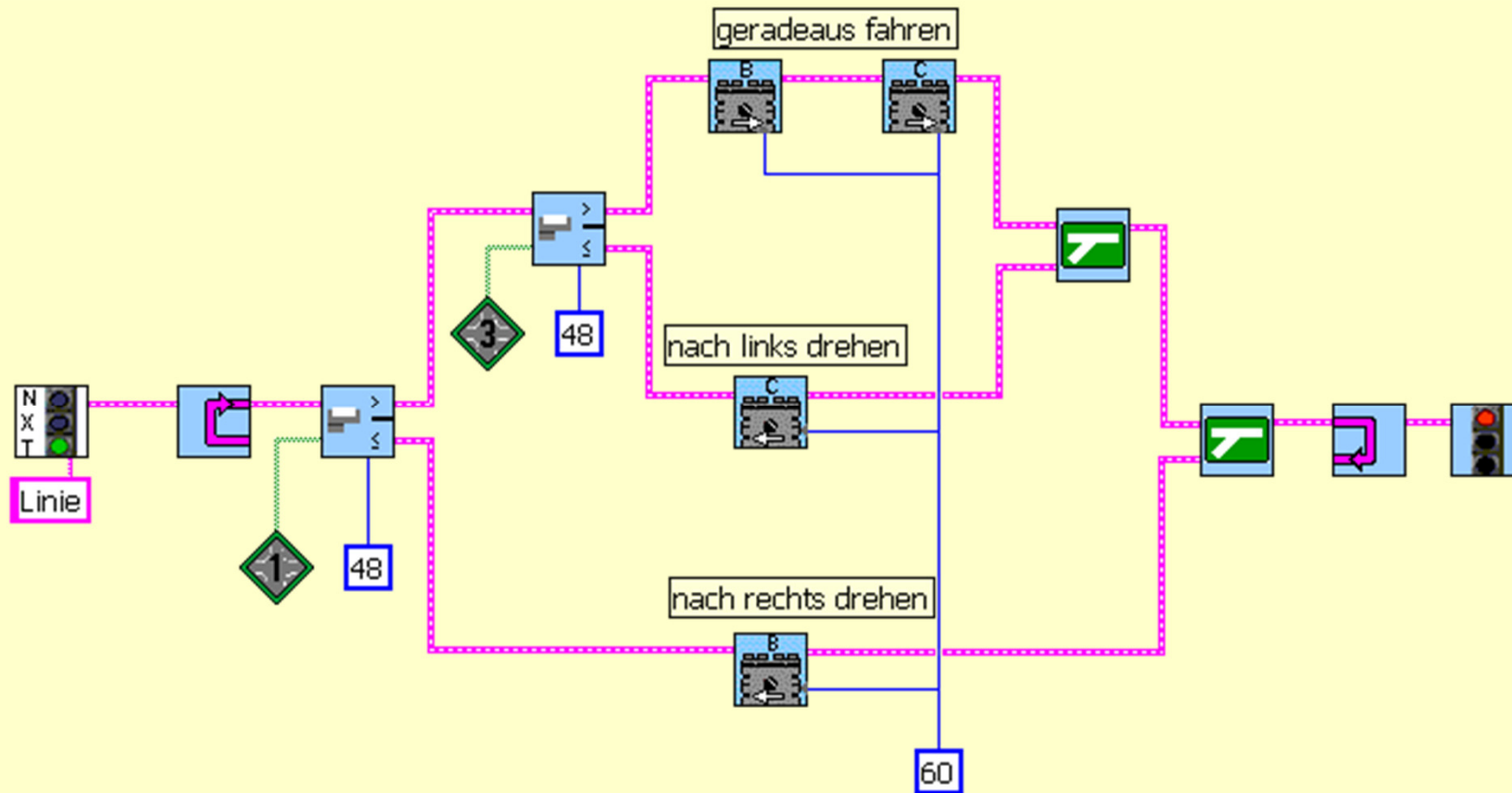


# Der Lösungsweg



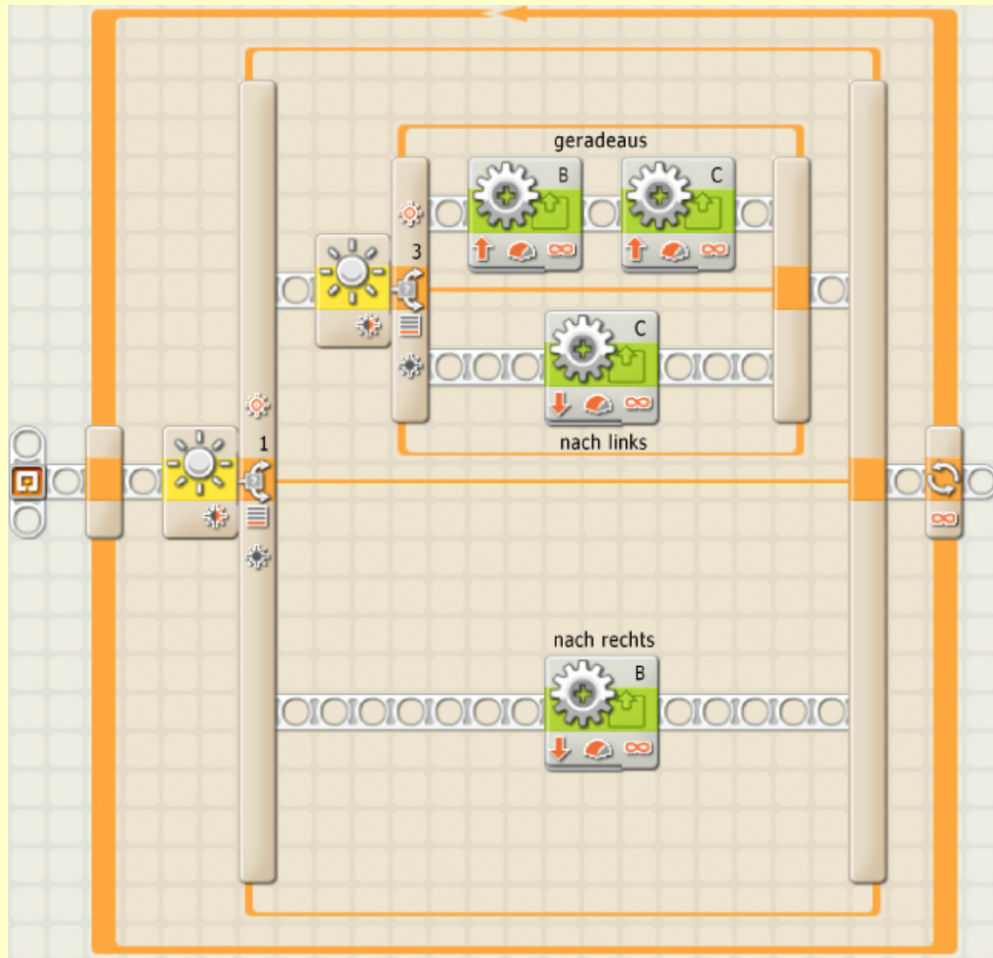


# Implementierung





# Implementierung





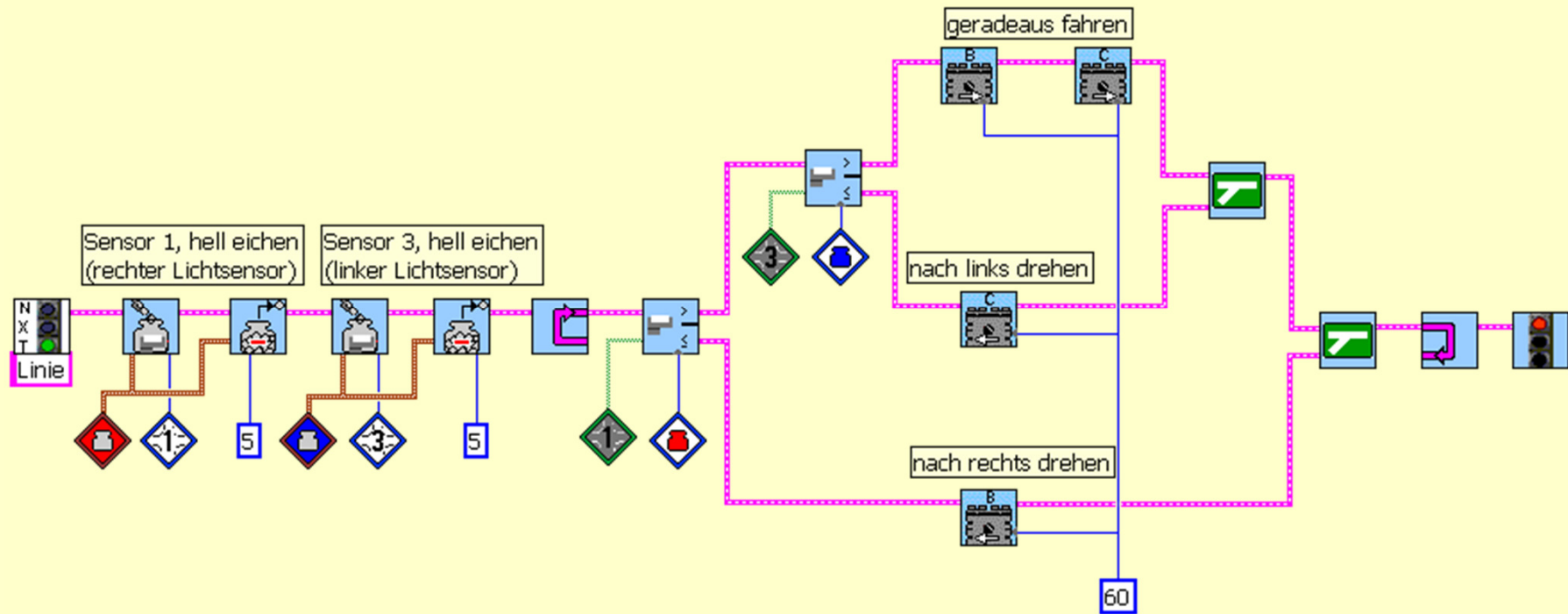
## *Fortsetzung*

- Eichen der beiden Sensoren
- Schwarzer Linie mit zwei Sensoren folgen mit Hilfe von zwei Tasks
- Schwarzer Linie auf heller Oberfläche und weisser Linie auf dunkler Oberfläche folgen
- Schwarzer Linie mit einem Sensor folgen



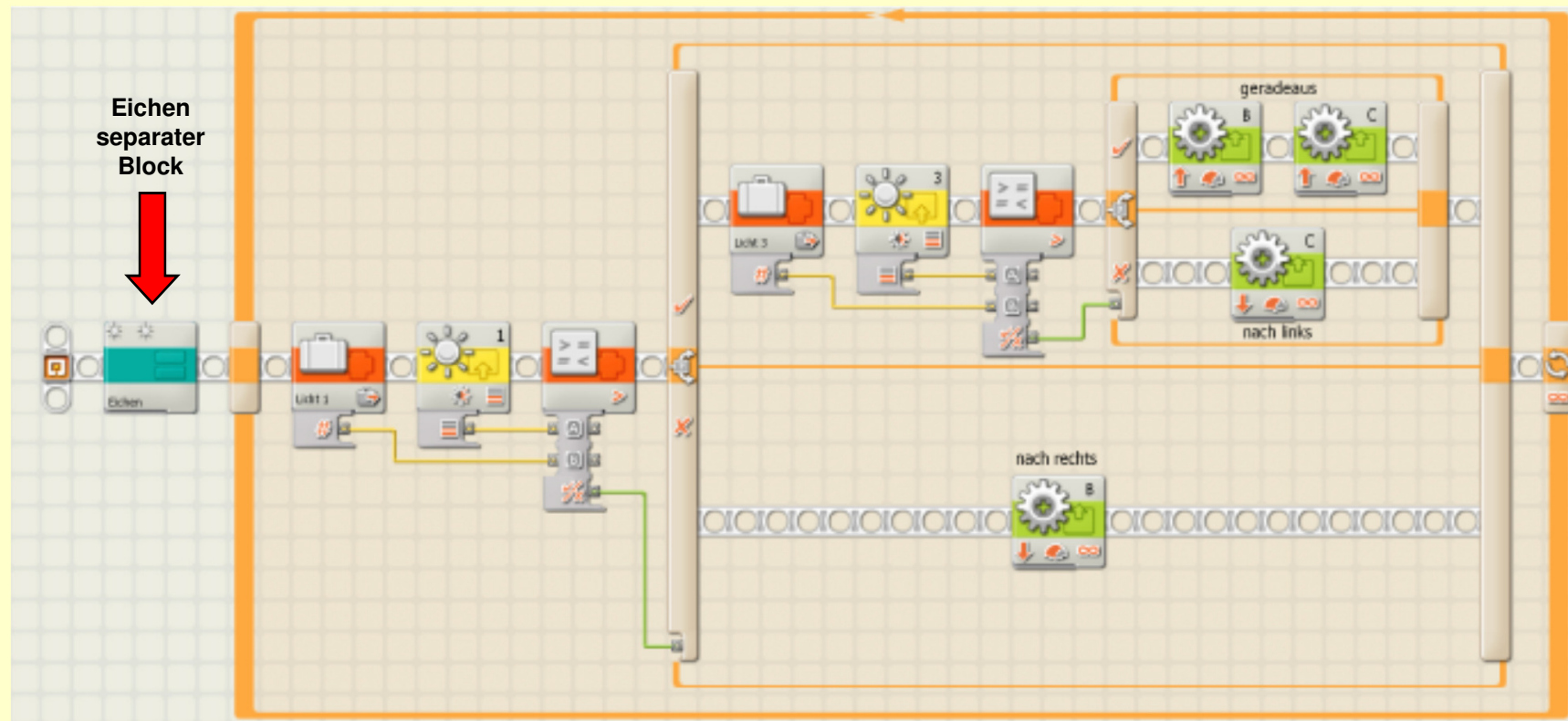


# Implementierung (mit Eichen)





# Implementierung *(Eichen separater Block)*



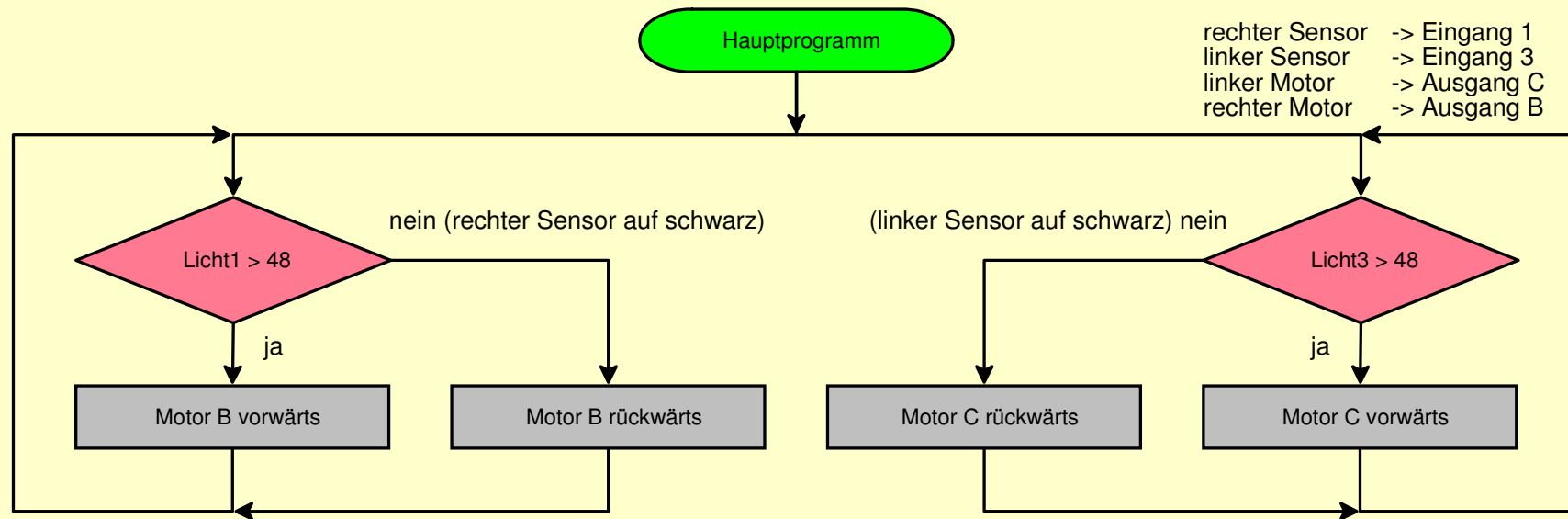


# Implementierung *(nur Eichen)*



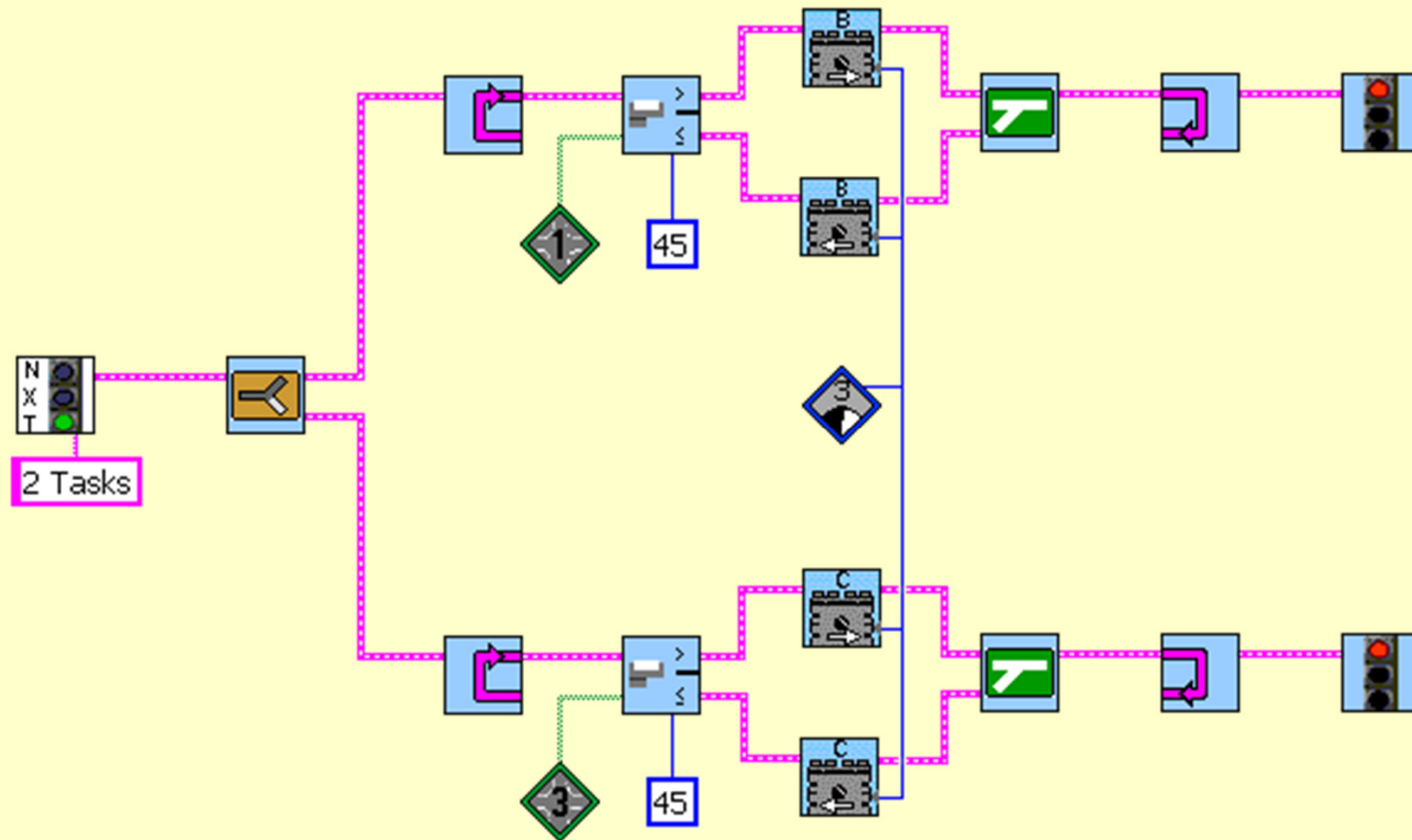


# Lösung mit zwei Tasks





# Implementierung





# Implementierung

