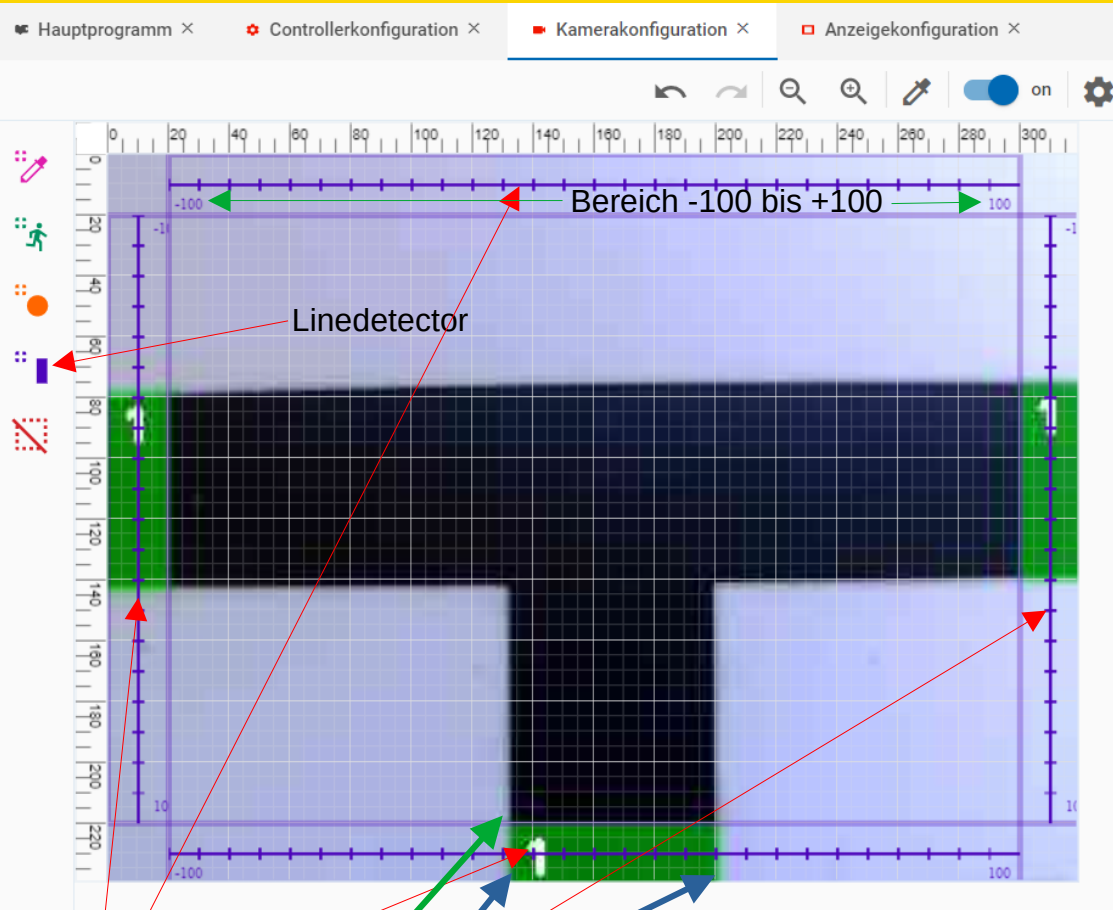


**Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Fischertechnik TXT 4.0
Teil 2.6: Linie**



Inspektor

Größe

Breite : px

Höhe : px

Position

x : px

y : px

Anzahl der Linien

Anzahl :

Linienbreite

Minimum :

Maximum :

Bereich

Start :

Ende :

Inspektor

Größe

Breite : px

Höhe : px

Position

x : px

y : px

Anzahl der Linien

Anzahl :

Linienbreite

Minimum :

Maximum :

Bereich

Start :

Ende :

Inspektor

Größe

Breite : px

Höhe : px

Position

x : px

y : px

Anzahl der Linien

Anzahl :

Linienbreite

Minimum :

Maximum :

Bereich

Start :

Ende :

Inspektor

Größe

Breite : px

Höhe : px

Position

x : px

y : px

Anzahl der Linien

Anzahl :

Linienbreite

Minimum :

Maximum :

Bereich

Start :

Ende :

line detector

line detector2

line detector22

line detector222

Die Namen lassen sich leider nicht ändern!

Die geeignetsten Parameter heraus finden und eintragen.
Das sind nicht notwendigerweise die abgebildeten!!!

LINIENERKENNUNG						
Name	Linie	Position	Breite	Rot	Grün	Blau
line_detector	1	4	49	10	0	0
line_detector2	1					
line_detector22	1	-10	66	11	8	6
line_detector222	1	-13	65	63	31	20

Anteile von Rot, Grün und Blau an der Farbe Schwarz

Wenn eine Linie einmal erkannt wurde, bleibt sie für immer im Gedächtnis des Roboters :(

Wenn eine Linie erkannt wurde, dann nehmen wir gleichzeitig die System-Zeit auf.

Übersteigt die System-Zeit die abgespeicherter Zeit um einen bestimmten Wert, löschen wir das Gedächtnis.

Ist umständlich, aber bisher gibt es kaum eine Alternative.

```
wenn Linien line_detector erkannt: Ereignisliste
  setze Position_Linie_vorne auf hole Position der Linie 1 aus Ereignis Liste
  setze Linie_vorne_Zeit auf Zeitstempel ms

wenn Linien line_detector2 erkannt: Ereignisliste
  setze Position_Linie_hinten auf hole Position der Linie 1 aus Ereignis Liste
  setze Linie_hinten_Zeit auf Zeitstempel ms

wenn Linien line_detector22 erkannt: Ereignisliste
  setze Position_Linie_links auf hole Position der Linie 1 aus Ereignis Liste
  setze Linie_links_Zeit auf Zeitstempel ms

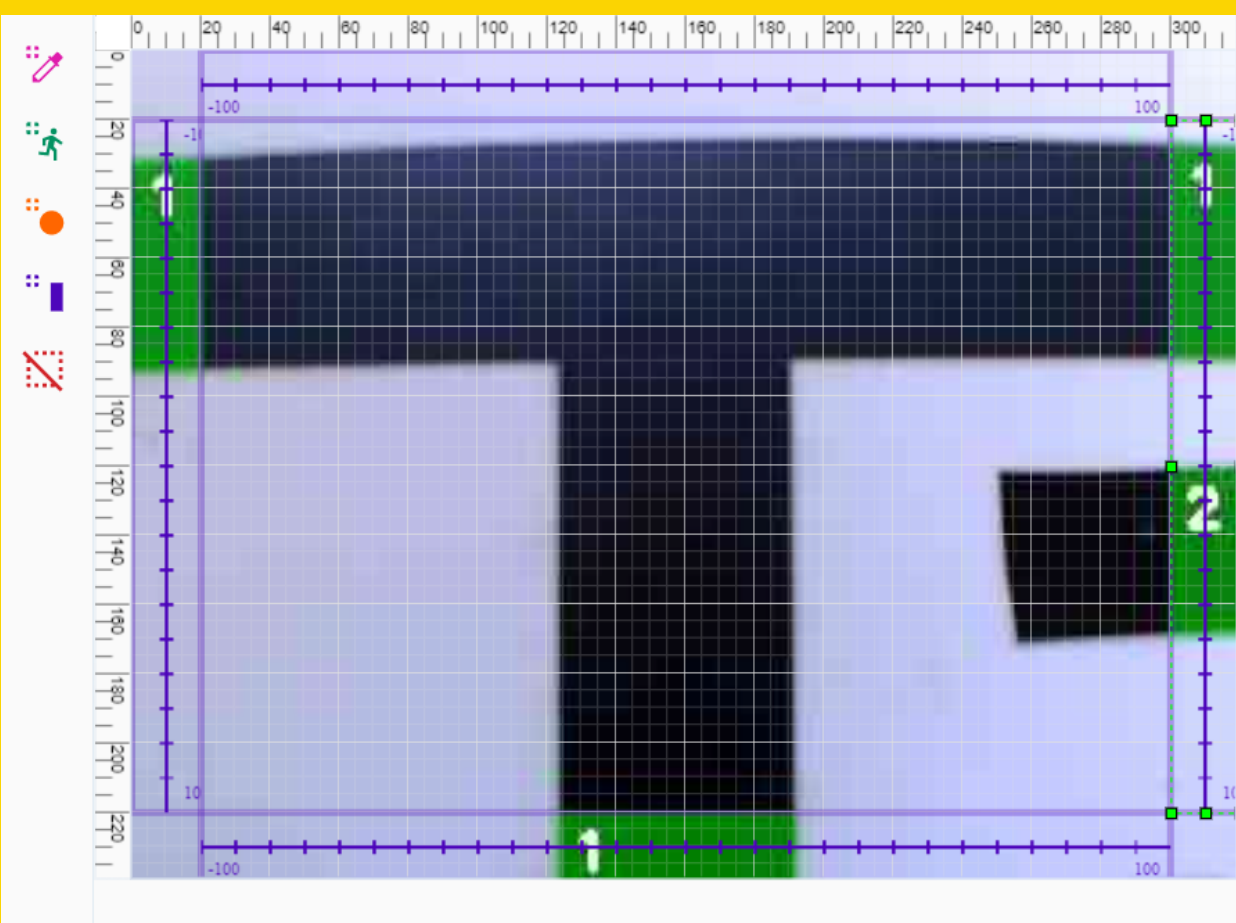
wenn Linien line_detector222 erkannt: Ereignisliste
  setze Position_Linie_rechts auf hole Position der Linie 1 aus Ereignis Liste
  setze Linie_rechts_Zeit auf Zeitstempel ms
```

```
+ um Linie_existiert?
  + falls Linie_vorne_Zeit + 80 < Zeitstempel ms
    mache setze Position_Linie_vorne auf 255
  + falls Linie_hinten_Zeit + 80 < Zeitstempel ms
    mache setze Position_Linie_hinten auf 255
  + falls Linie_links_Zeit + 80 < Zeitstempel ms
    mache setze Position_Linie_links auf 255
  + falls Linie_rechts_Zeit + 80 < Zeitstempel ms
    mache setze Position_Linie_rechts auf 255
```

Der Wert 80 scheint ganz gut zu funktionieren, aber darauf verlassen, solltet ihr euch nicht!
Warum nehmen wir den Wert 255?
Warum nicht 0?

Leider müssen wir im Hauptprogramm dafür sorgen, dass die Funktion „Linie existiert?“ immer wieder ausgeführt wird. Auch in jeder Funktion, die länger als 80ms (Beispiel) dauert und in der wir die Linienerkennung brauchen.

```
Programmstart
Anfangswerte
wiederhole bis  ist Kontrollkästchen taster checked
  mache Display
  Display Linie_existiert?
  setze Statusanzeige status aktiv wahr
dauerhaft wiederholen
  mache Display
  Display Linie_existiert?
  forward
  warte s 1
  stop
  warte s 10
```



Größe

Breite : px

Höhe : px

Position

x : px

y : px

Anzahl der Linien

Anzahl :

Linienbreite

Minimum :

Maximum :

Bereich

Start :

Ende :

Es gibt die Möglichkeit, mehr als eine Linie pro line_detector zu erkennen. Dazu muss der Parameter „Anzahl der Linien“ entsprechend verändert werden. Die Auswertung ist die gleiche wie vorher. Es muss dann der Wert von Linie „2“ ausgewertet werden.

```
wenn Linien line_detector222 erkannt: Ereignisliste
  setze Position_Linie_rechts auf hole Position der Linie 2 aus Ereignis Liste
  setze Linie_rechts_Zeit auf Zeitstempel ms
```

Konsole	Kamera					
line_detector222	1	-62	61	58	35	27
	2	24	48	21	10	10

Haltepunkte

Vielleicht erweitere ich das Tutorial nächstes Jahr. Dieses Jahr konzentriere ich mich jedoch auf Maze Entry...

Viel Erfolg!

Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Fischertechnik TXT 4.0
Teil 2.7: Gibt es noch nicht...