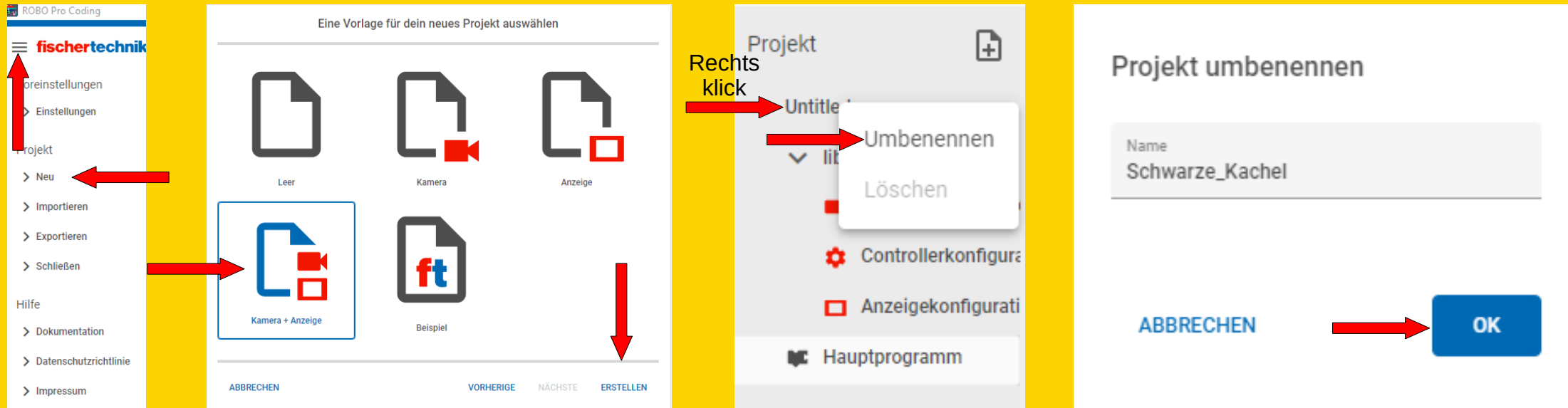


Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Fischertechnik TXT 4.0
Teil 2.3: Schwarze Kachel

Jetzt wird es ein wenig ärgerlich: Wir müssen ein neues Projekt erstellen, in dem wir zusätzlich die Kamera benutzen wollen. Leider können wir nicht copy and paste benutzen. Wir müssen alles nochmal eingeben. Schwacher Trost: Wiederholung übt!



Und damit es nicht gar zu schwer wird, kommen jetzt Bilder vom letzten Programm:

```

+ um Anfangswerte
  setze Ultraschall_Wert auf 0
  setze Geschwindigkeit auf 384
  setze 30cm auf 219
  setze 90Grad auf 121
  setze Zu_Nah_Vorne auf 1500
  setze Zu_Fern_Vorne auf 1000
  setze Keine_Wand auf 600
  setze Strecke_Wand_Ende auf 110
  setze Statusanzeige kontrolle aktiv falsch
  setze Kontrollkästchen taster checked falsch
  setze Beschriftungsfeld us_text Text "Ultraschall"
  setze Beschriftungsfeld lab_rechts_vorne Text "Rechts Vorne"
  setze Beschriftungsfeld lab_rechts_hinten Text "Rechts Hinten"
  T setze Eingabefeld us_wert Text "0"
  T setze Eingabefeld in_rechts_vorne Text "0"
  T setze Eingabefeld in_rechts_hinten Text "0"

```

Hauptprogramm

Ultraschall	0	
Rechts Vorne	0	
Rechts Hinten	0	

Anzeige-Konfiguration

onoff

```

Programmstart
Anfangswerte
wiederhole bis ist Kontrollkästchen taster checked
mache Werte_auf_Display
  setze Statusanzeige kontrolle aktiv wahr
dauerhaft wiederholen
mache Werte_auf_Display
  + falls ist Ultraschallsensor TXT_M_18 Abstand > 6
    mache + falls hole Farbsensor TXT_M_17 Wert > Keine_Wand
      mache + falls hole Farbsensor TXT_M_17 Wert > Zu_Nah_Vorne
        mache
          setze Statusanzeige korrektur_links aktiv wahr
          setze Statusanzeige korrektur_rechts aktiv falsch
          Korrektur_links
        sonst + falls hole Farbsensor TXT_M_17 Wert < Zu_Fern_Vorne
          mache
            setze Statusanzeige korrektur_links aktiv falsch
            setze Statusanzeige korrektur_rechts aktiv wahr
            Korrektur_rechts
          sonst
            setze Statusanzeige korrektur_rechts aktiv falsch
            setze Statusanzeige korrektur_links aktiv falsch
            Forward
        sonst
          Variables_Drehen_Links mit:
            Velocity Geschwindigkeit
            Distance 90Grad
            wiederhole bis hat Motor TXT_M_M1 Position erreicht
            mache Werte_auf_Display
      sonst
        Variables_Drehen_Rechts mit:
          Velocity Geschwindigkeit
          Distance 90Grad
          wiederhole bis hat Motor TXT_M_M1 Position erreicht
          mache Werte_auf_Display
    sonst
      Variable_Strecke_Vorwärts mit:
        Velocity Geschwindigkeit
        Distance Strecke_Wand_Ende
        wiederhole bis hat Motor TXT_M_M1 Position erreicht
        mache Werte_auf_Display
      Variable_Strecke_Rückwärts mit:
        Velocity Geschwindigkeit
        Distance Strecke_Wand_Ende
        wiederhole bis hat Motor TXT_M_M1 Position erreicht
        mache Werte_auf_Display

```

Hauptprogramm

```

+ um Forward
  + setze Motor TXT_M_M1 links Geschwindigkeit 512
  sync mit Motor TXT_M_M2 Richtung links
  sync mit Motor TXT_M_M3 Richtung links
  sync mit Motor TXT_M_M4 Richtung links

+ um Backward
  + setze Motor TXT_M_M1 rechts Geschwindigkeit 512
  sync mit Motor TXT_M_M2 Richtung rechts
  sync mit Motor TXT_M_M3 Richtung rechts
  sync mit Motor TXT_M_M4 Richtung rechts

+ um Turn_Right
  + setze Motor TXT_M_M1 rechts Geschwindigkeit 512
  sync mit Motor TXT_M_M2 Richtung links
  sync mit Motor TXT_M_M3 Richtung rechts
  sync mit Motor TXT_M_M4 Richtung links

+ um Turn_Left
  + setze Motor TXT_M_M1 links Geschwindigkeit 512
  sync mit Motor TXT_M_M2 Richtung rechts
  sync mit Motor TXT_M_M3 Richtung links
  sync mit Motor TXT_M_M4 Richtung rechts

+ um Stop
  + stoppe Motor TXT_M_M1
  sync mit Motor TXT_M_M2
  sync mit Motor TXT_M_M3
  sync mit Motor TXT_M_M4

```

Funktionen

```

+ um Variables_Drehen_Links mit:
  - variable Velocity
  - variable Distance
  + setze Motor TXT_M_M1 links Geschwindigkeit Velocity
  Schrittweite Distance
  sync mit Motor TXT_M_M2 Richtung rechts
  sync mit Motor TXT_M_M3 Richtung links
  sync mit Motor TXT_M_M4 Richtung rechts

+ um Ende_Wand
  Variable_Strecke_Vorwärts mit:
    Velocity Geschwindigkeit
    Distance Strecke_Wand_Ende
  wiederhole bis hat Motor TXT_M_M1 Position erreicht
  mache Werte_auf_Display
  Variables_Drehen_Rechts mit:
    Velocity Geschwindigkeit
    Distance 90Grad
  wiederhole bis hat Motor TXT_M_M1 Position erreicht
  mache Werte_auf_Display
  Variable_Strecke_Vorwärts mit:
    Velocity Geschwindigkeit
    Distance Strecke_Wand_Ende
  wiederhole bis hat Motor TXT_M_M1 Position erreicht
  mache Werte_auf_Display

```

Motor

```

+ um Trip_Forward
  + setze Motor TXT_M_M1 links Geschwindigkeit 512
  Schrittweite 100
  sync mit Motor TXT_M_M2 Richtung links
  sync mit Motor TXT_M_M3 Richtung links
  sync mit Motor TXT_M_M4 Richtung links

+ um Trip_Backward
  + setze Motor TXT_M_M1 rechts Geschwindigkeit 512
  Schrittweite 100
  sync mit Motor TXT_M_M2 Richtung rechts
  sync mit Motor TXT_M_M3 Richtung rechts
  sync mit Motor TXT_M_M4 Richtung rechts

+ um Trip_Turn_Right
  + setze Motor TXT_M_M1 rechts Geschwindigkeit 512
  Schrittweite 100
  sync mit Motor TXT_M_M2 Richtung links
  sync mit Motor TXT_M_M3 Richtung rechts
  sync mit Motor TXT_M_M4 Richtung links

+ um Trip_Turn_Left
  + setze Motor TXT_M_M1 links Geschwindigkeit 512
  Schrittweite 100
  sync mit Motor TXT_M_M2 Richtung rechts
  sync mit Motor TXT_M_M3 Richtung links
  sync mit Motor TXT_M_M4 Richtung rechts

```

```

+ um Korrektur_rechts
  + setze Motor TXT_M_M1 links Geschwindigkeit 384
  Schrittweite 480
  + setze Motor TXT_M_M2 links Geschwindigkeit 480
  sync mit Motor TXT_M_M4 Richtung links

+ um Korrektur_links
  + setze Motor TXT_M_M1 links Geschwindigkeit 480
  Schrittweite 384
  + setze Motor TXT_M_M2 links Geschwindigkeit 384
  sync mit Motor TXT_M_M4 Richtung links

```

Anzeige

```

+ um Variable_Strecke_Vorwärts mit:
  - variable Velocity
  - variable Distance
  + setze Motor TXT_M_M1 links Geschwindigkeit Velocity
  Schrittweite Distance
  sync mit Motor TXT_M_M2 Richtung links
  sync mit Motor TXT_M_M3 Richtung links
  sync mit Motor TXT_M_M4 Richtung links

+ um Variable_Strecke_Rückwärts mit:
  - variable Velocity
  - variable Distance
  + setze Motor TXT_M_M1 rechts Geschwindigkeit Velocity
  Schrittweite Distance
  sync mit Motor TXT_M_M2 Richtung rechts
  sync mit Motor TXT_M_M3 Richtung rechts
  sync mit Motor TXT_M_M4 Richtung rechts

+ um Variables_Drehen_Rechts mit:
  - variable Velocity
  - variable Distance
  + setze Motor TXT_M_M1 rechts Geschwindigkeit Velocity
  Schrittweite Distance
  sync mit Motor TXT_M_M2 Richtung links
  sync mit Motor TXT_M_M3 Richtung rechts
  sync mit Motor TXT_M_M4 Richtung links

```

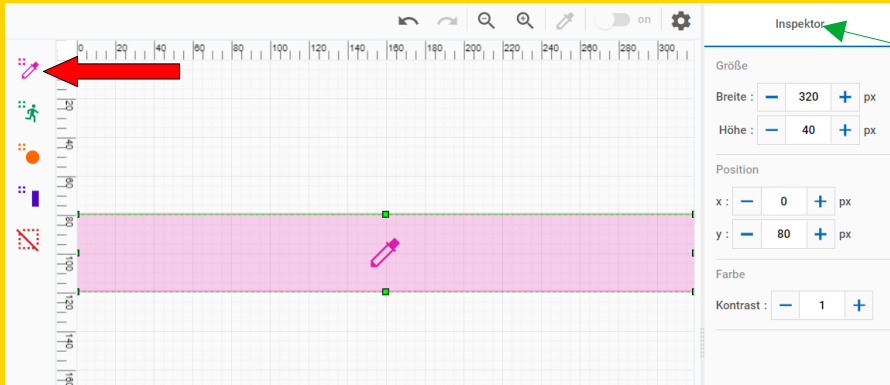
```

+ um Werte_auf_Display
  T setze Eingabefeld us_wert Text
  T setze Eingabefeld in_rechts_vorne Text
  T setze Eingabefeld in_rechts_hinten Text

```

Kamerakonfiguration

Um die schwarze Kachel zu erkennen, benutzen wir die Kamera. color_detector halten und auf die Fläche ziehen.



Mit dem Inspektor den Bereich bestimmen, in dem wir die schwarze Kachel erkennen wollen.

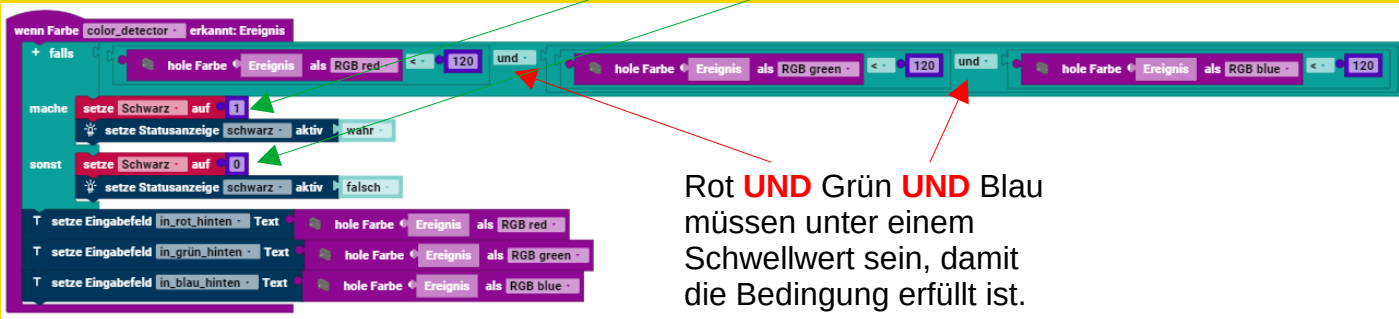


Der Roboter steht in der Mitte des Feldes vor der schwarzen Kachel. (Grenze grün) Mit der Kamera sehen wir schon das schwarze Feld. Wir müssen also erst garnicht drauf fahren, sondern können gleich ausweichen.

USB

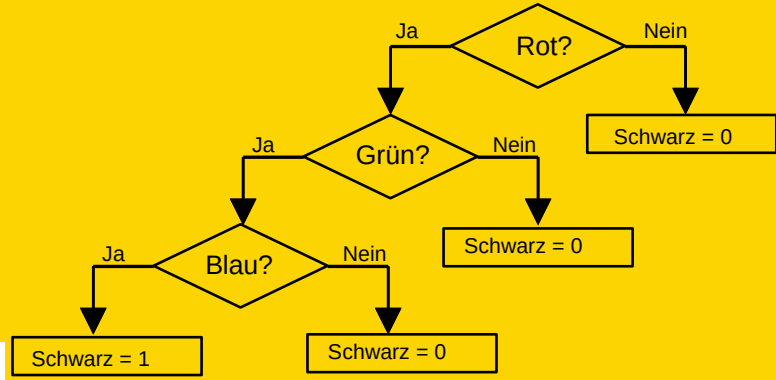


Die Farbe können wir nicht einfach im Hauptprogramm abfragen. Wir müssen den Umweg über „wenn Farbe ... erkannt: Ereignis“ gehen. Darin entscheiden wir, ob wir schwarz sehen (1) oder nicht (0).



Rot **UND** Grün **UND** Blau müssen unter einem Schwellwert sein, damit die Bedingung erfüllt ist.

Baumdiagramm



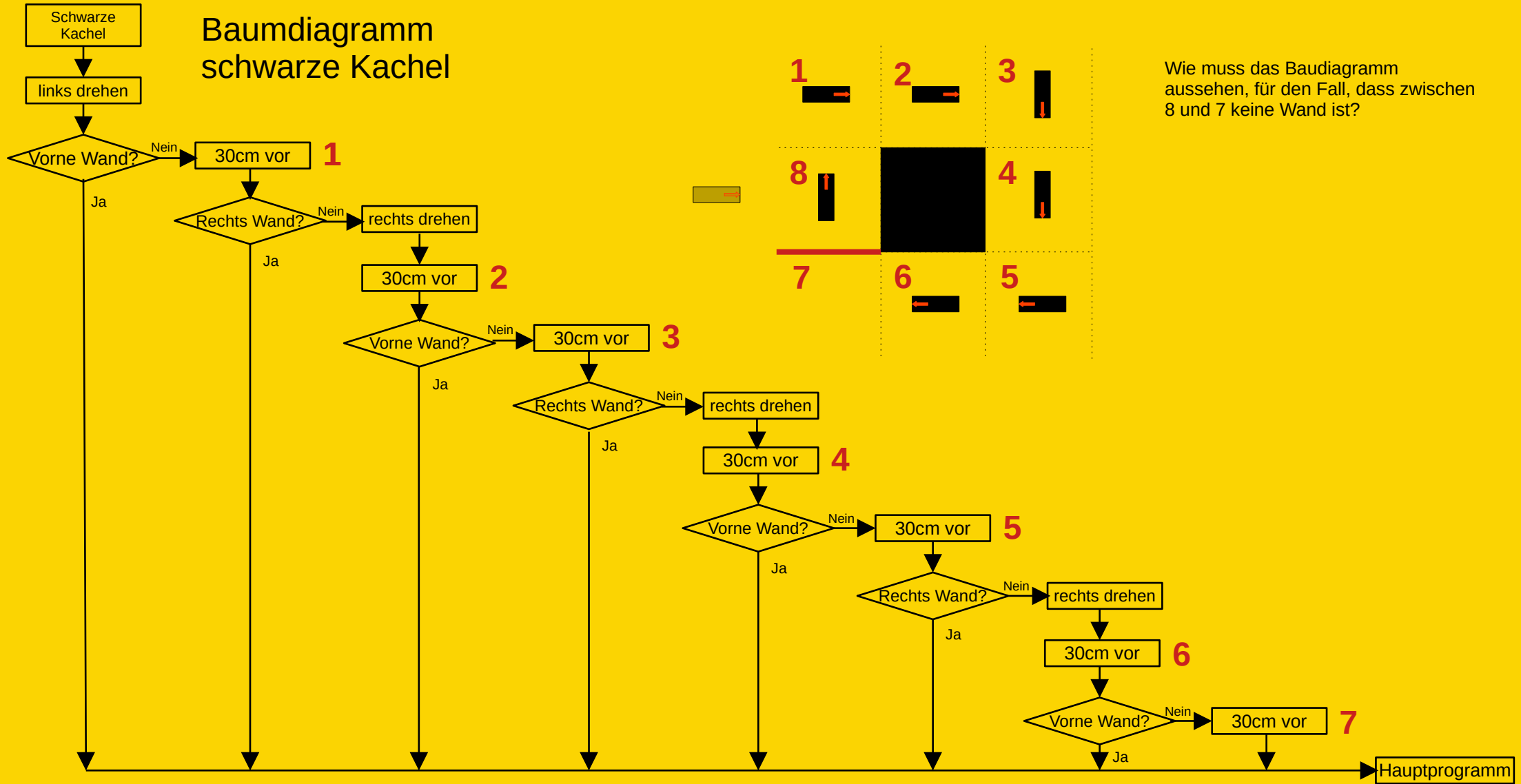
Rot **UND** Grün **UND** Blau liegen unter dem Schwellwert

Wir holen uns alle Farbwerte im RGB-Format, Rot, Grün, Blau mit Werten zwischen 0 und 255. Schaut euch die Werte für einen weißen Untergrund und für einen schwarzen Untergrund an. Weiß erzeugt hohe Werte bei allen Farben. Schwarz erzeugt kleine Werte bei allen Farben. Welche Werte erhaltet ihr? Bildet den Mitten-Wert (NICHT Mittelwert!). Oberhalb ist Weiß, Unterhalb ist Schwarz.

Die Werte müssen vor jedem Start womöglich angepasst werden. Ihr müsst euren „Sensor“ kalibrieren.

Aufgabe: Erstellt ein Baumdiagramm für den Fall, dass der Roboter auf eine schwarze Fläche trifft. Wie muss der Roboter weiter fahren? Was müsst ihr immer wieder überprüfen? Erstellt dazu neue Funktionen: 30cm vorwärts, 90 Grad Drehung links und rechts. Packt die „Schwarze Platte“ in eine Funktion. An welche Stelle kommt die Funktion ins Hauptprogramm?

Baumdiagramm schwarze Kachel



Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Fischertechnik TXT 4.0
Teil 2.5: Rotes Kreuz