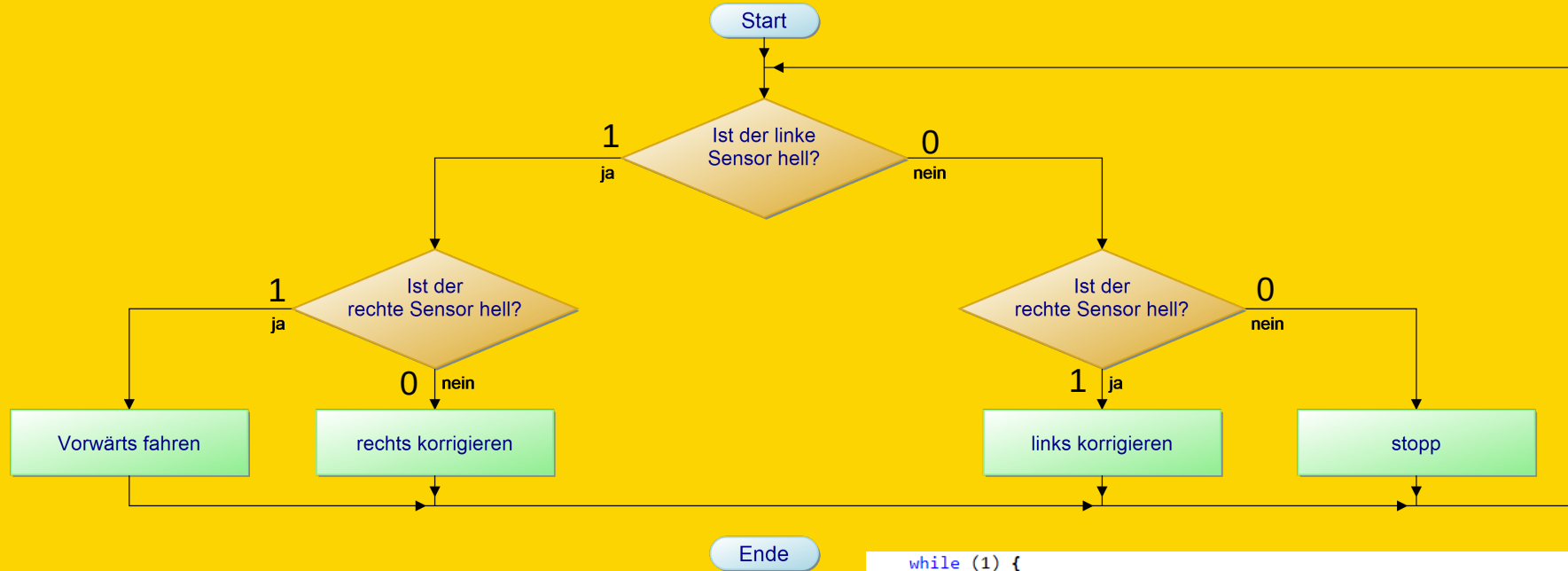


Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Arduino Nano
Teil 2.7: Line Follower

Von Charlotte und Andreas

Linienfolger



Ende

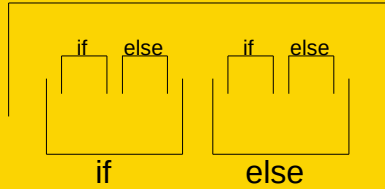
Wahrheitstabelle zum Baumdiagramm:

Linker Sensor		Rechter Sensor	Aktion
Dunkel	= 0	Dunkel = 0	Stop()
Dunkel	= 0	Hell = 1	Turn_left()
Hell	= 1	Dunkel = 0	Turn_right()
Hell	= 1	Hell = 1	Forward()

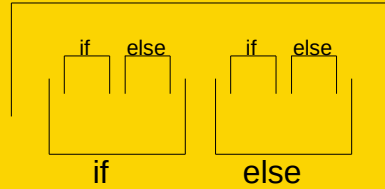
```
while (1) {
    Data_Visualizer(); //Define the values to be displayed and transmit them
    if (BlacknWhite_left) { //Left light sensor bright?
        if (BlacknWhite_right) { //Right light sensor bright?
            Forward(128,128); //With half force
        }
        else { //Right light sensor dark
            Turn_left(128); //Turn on the spot with half force
        }
    }
    else { //Left light sensor dark
        if (BlacknWhite_right) { //Right light sensor bright?
            Turn_right(128); //Turn on the spot
        }
        else { //Both sensors are dark
            Stop(); //For the moment
        }
    }
}
```

Schachtelung

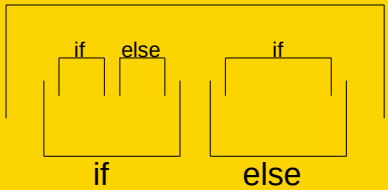
if



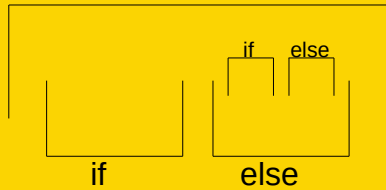
else



if



else



Äußere Struktur

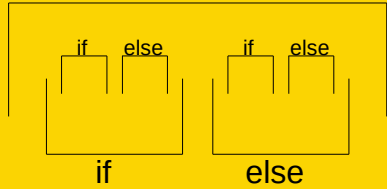
Innere Struktur

Innere Struktur

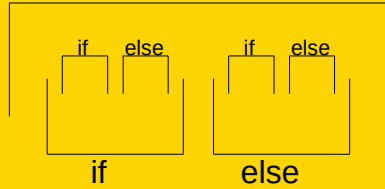
```
while (1) { //1 turn on, 0 turn off
    Data_Visualizer(); //Define the values to be displayed and transmit them
    if (BlacknWhite_left) { //Left light sensor bright?
        if (BlacknWhite_right) { //Right light sensor bright?
            Forward(128,128); //With half force
        }
        else { //Right light sensor dark
            Turn_left(128); //Turn on the spot with half force
        }
    }
    else { //Left light sensor dark
        if (BlacknWhite_right) { //Right light sensor bright?
            Turn_right(128); //Turn on the spot
        }
        else { //Both sensors are dark
            Stop(); //For the moment
        }
    }
}
```

Schachtelung

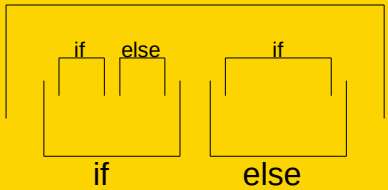
if



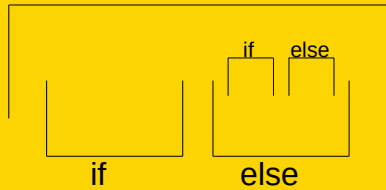
else



if



else



Äußere Struktur

Innere Struktur

Innere Struktur

Nicht eingerückt:
Was gehört hier
zu was?

```
if (Line_digital_left) { //Left light sensor bright?
if (Line_digital_right) { //Right light sensor bright?
  Forward(128,128); //With half force
}
else { //Right light sensor dark
  Turn_right(128); //Turn on the spot
}
}
else { //Left light sensor dark
if (Line_digital_right) { //Right light sensor bright?
  Turn_left(128); //Turn on the spot
}
}
else { //Both sensors are dark
  Stop(); //For the moment
}
}
```

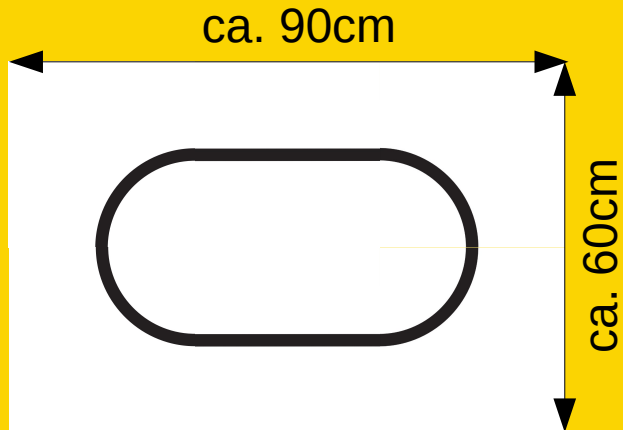
```
while (1) { //1 turn on, 0 turn off
  Data_Visualizer(); //Define the values to be displayed and transmit them
  if (BlacknWhite_left) { //Left light sensor bright?
    if (BlacknWhite_right) { //Right light sensor bright?
      Forward(128,128); //With half force
    }
    else { //Right light sensor dark
      Turn_left(128); //Turn on the spot with half force
    }
  }
  else { //Left light sensor dark
    if (BlacknWhite_right) { //Right light sensor bright?
      Turn_right(128); //Turn on the spot
    }
    else { //Both sensors are dark
      Stop(); //For the moment
    }
  }
}
```

Aufgabe 1:

Baut euch einen einfachen Parcours auf und testet das Programm.

Mit welcher Geschwindigkeit schafft der Roboter noch die Kurven.

(Solltet ihr keine Kacheln haben, dann Benutzt schwarzes Isolierband, ca. 15mm breit)



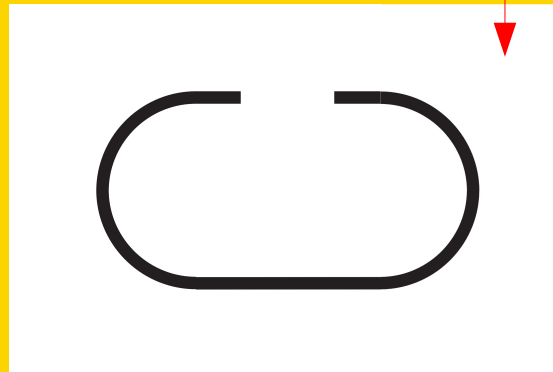
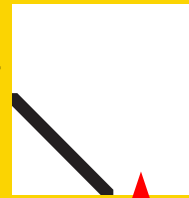
Aufgabe 2:

Baut in den Parcours eine Lücke (Gap) ein und testet das Programm.

Was verändert sich?.

(Solltet ihr keine Kacheln haben, dann schneidet das Isolierband, ca. 20cm aus.)

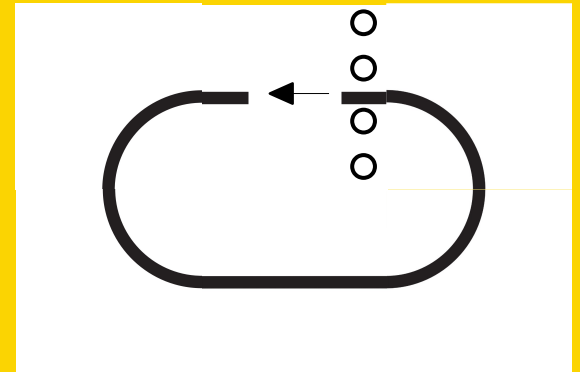
Tauscht eine Kurven-Kachel gegen eine Diagonale aus.



Aufgabe 3:

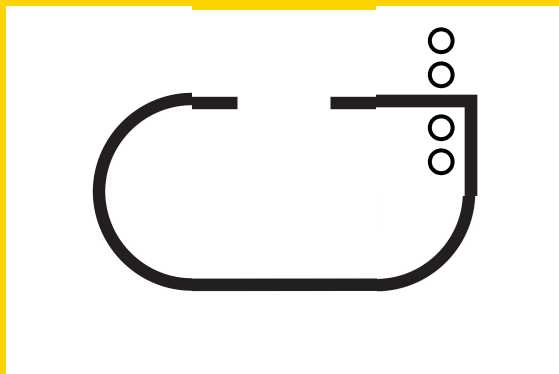
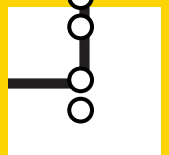
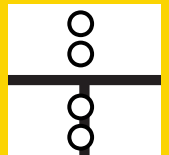
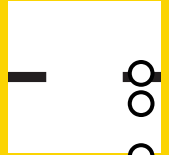
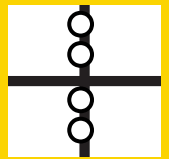
Baut in das Programm zusätzliche Sensoren mit ein.

Macht zunächst ein Baumdiagramm oder eine Wahrheitstabelle Und überlegt euch, welche Reaktion bei welchen Zuständen der Sensoren der Roboter ausführen soll.



Wahrheitstabelle vier Sensoren: (hell = 1, dunkel = 0)

Linker Sensor	Mitte links Sensor	Mitte rechts Sensor	Rechter Sensor	Situation
0	0	0	0	(0) Kreuzung?
0	0	0	1	(1) Abknicken nach links
0	0	1	0	(2) Geht das?
0	0	1	1	(3) Abknicken nach rechts
0	1	0	0	(4) Geht das?
0	1	0	1	(5) Geht das?
0	1	1	0	(6) Geht das?
0	1	1	1	(7) Nach Gap, Linie wieder gefunden
1	0	0	0	(8) ...



Abknickung folgen: Noch ein Stück weiterfahren und um den Mittelpunkt des Roboters solange (while) drehen, bis Linie wieder erreicht ist.

Wie unterscheiden wir eine Abknickung von einer T-Kreuzung?

Welche Merkmale hat eine Kreuzung zusätzlich zur Abknickung?

Wieviele Encoder-Ticks vorwärts fahren entspricht welcher Strecke?
(Lasst den Roboter 200 Encoder-Ticks fahren und misst die Strecke. Dann rechnet Teilstrecken aus...)

Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Elegoo Car Kit
Teil 2.8: Obstacle

Von Charlotte und Andreas