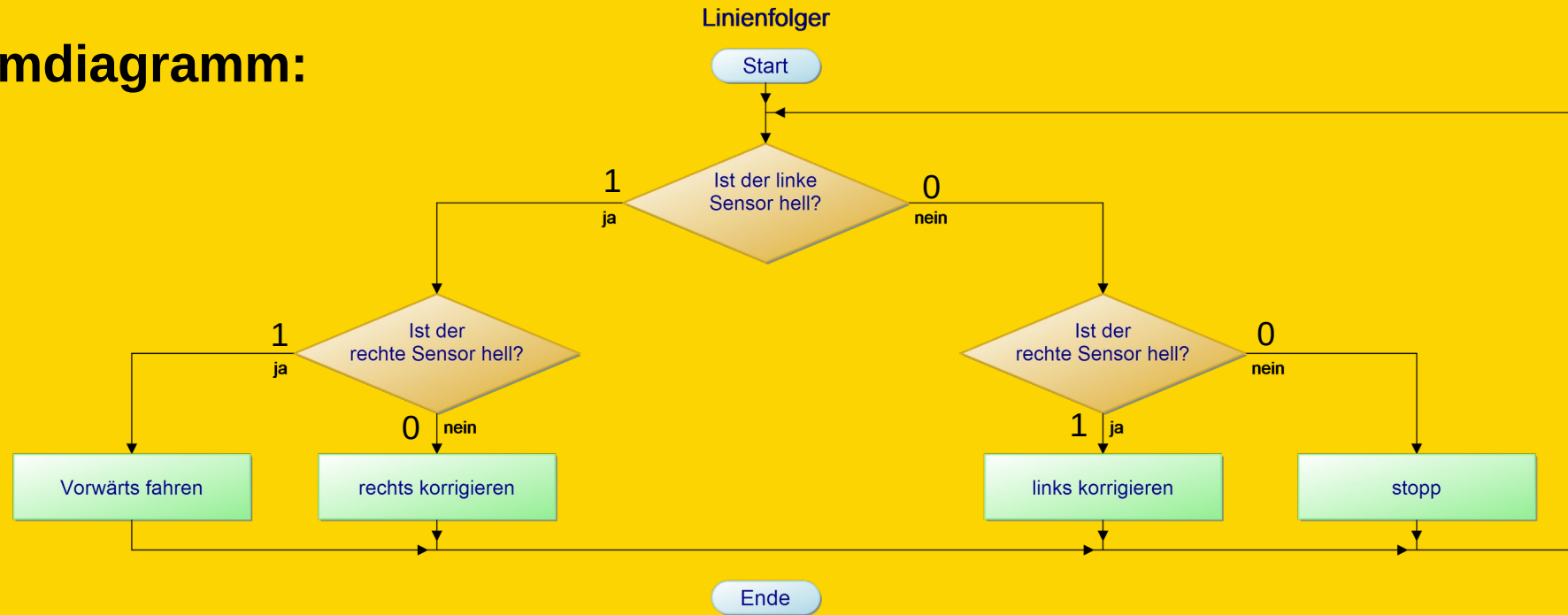


Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Elegoo Car Kit
Teil 2.7: Line Follower

Von Charlotte und Andreas

Baumdiagramm:



Wahrheitstabelle zum Baumdiagramm:

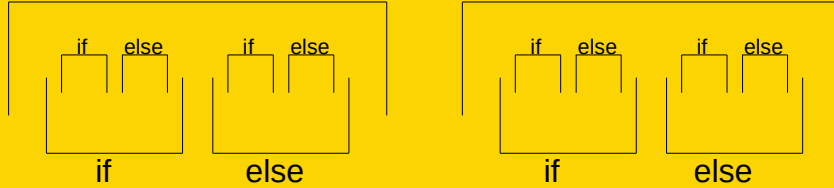
Linker Sensor		Rechter Sensor		Aktion
Dunkel	= 0	Dunkel	= 0	Stop()
Dunkel	= 0	Hell	= 1	Turn_left()
Hell	= 1	Dunkel	= 0	Turn_right()
Hell	= 1	Hell	= 1	Forward()

```
85 while (1) {
86     Data_Visualizer(); //Define the values to be displayed and transmit them
87     if (Line_digital_left) { //Left light sensor bright?
88         if (Line_digital_right) { //Right light sensor bright?
89             Forward(128,128); //With half force
90         }
91         else { //Right light sensor dark
92             Turn_right(128); //Turn on the spot
93         }
94     }
95     else { //Left light sensor dark
96         if (Line_digital_right) { //Right light sensor bright?
97             Turn_left(128); //Turn on the spot with half force
98         }
99         else { //Both sensors are dark
100             Stop(); //For the moment
101         }
102     }
103 }
```

Schachtelung

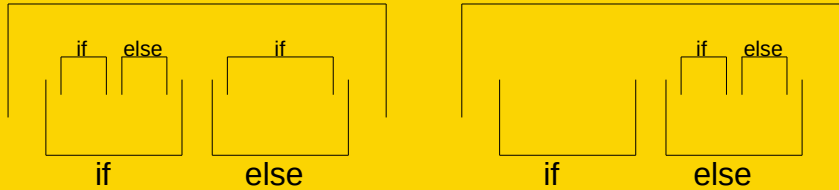
if

else



if

else



Äußere Struktur

Innere Struktur

Innere Struktur

.07_Line_Follower - AtmelStudio

File Edit View VAssistX ASF Project Build Debug Tools Window Help

Debug Debug Browser

Line.cpp USART_Functions.h Drive_Functions.h Interrupt_Service_Routines.h Inith

```
76
77 //Main routine
78 int main(void) {
79     Init(); //Initialize the MC
80     USART_Init(MYUBRR); //Transmission/reception initialize
81     sei(); //Enable all interrupts
82     //cli(); //Clear all interrupts
83     Stop();
84     _delay_ms(1000);
85     while (1) {
86         Data_Visualizer(); //Define the values to be displayed and transmit them
87         /* Hier die Abfrage für den dritten Sensor einfügen */
88         /* Hier die Abfrage für den dritten Sensor einfügen */
89         /* Hier die Abfrage für den dritten Sensor einfügen */
90         if (Line_digital_left) { //Left light sensor bright?
91             if (Line_digital_right) { //Right light sensor bright?
92                 Forward(128,128); //With half force
93             }
94             else { //Right light sensor dark
95                 Turn_right(128); //Turn on the spot
96             }
97         }
98         else { //Left light sensor dark
99             if (Line_digital_right) { //Right light sensor bright?
100                 Turn_left(128); //Turn on the spot with half force
101             }
102             else { //Both sensors are dark
103                 Stop(); //For the moment
104             }
105         }
106         /* Hier die Alternative (else) des dritten Sensors einfügen */
107         /* Was ist mit den beiden anderen Sensoren? */
108         /* Hier die Alternative (else) des dritten Sensors einfügen */
109         /* Was ist mit den beiden anderen Sensoren? */
110     }
111 }
```

100 %

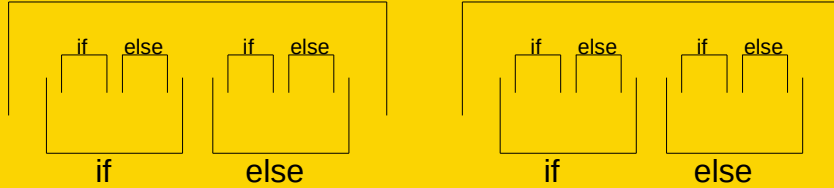
Output Error List

Item(s) Saved

Schachtelung

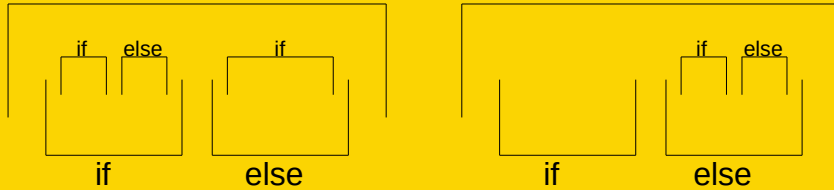
if

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 with half force
}
}
else { //Both sensors are dark
    Stop(); //For the moment
}
}
```

.07_Line_Follower - AtmelStudio

File Edit View VAssistX ASF Project Build Debug Tools Window Help

Line.cpp USART_Functions.h Drive_Functions.h Interrupt_Service_Routines.h Inith

```
76
77 //Main routine
78 int main(void) {
79     Init(); //Initialize the MC
80     USART_Init(MYUBRR); //Transmission/reception initialize
81     sei(); //Enable all interrupts
82     //cli(); //Clear all interrupts
83     Stop();
84     _delay_ms(1000);
85     while (1) {
86         Data_Visualizer(); //Define the values to be displayed and transmit them
87         /******
88         /* Hier die Abfrage für den dritten Sensor einfügen
89         /******
90         if (Line_digital_left) { //Left light sensor bright?
91             if (Line_digital_right) { //Right light sensor bright?
92                 Forward(128,128); //With half force
93             }
94             else { //Right light sensor dark
95                 Turn_right(128); //Turn on the spot
96             }
97         }
98         else { //Left light sensor dark
99             if (Line_digital_right) { //Right light sensor bright?
100                 Turn_left(128); //Turn on the spot with half force
101             }
102             else { //Both sensors are dark
103                 Stop(); //For the moment
104             }
105         }
106         /******
107         /* Hier die Alternative (else) des dritten Sensors einfügen
108         /* Was ist mit den beiden anderen Sensoren?
109         /******
110     }
```

Output Error List

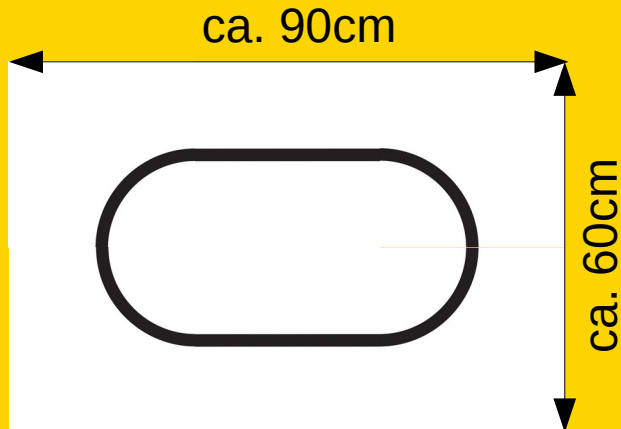
Item(s) Saved

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)

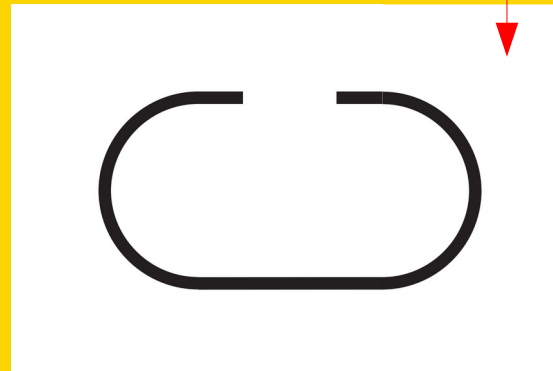
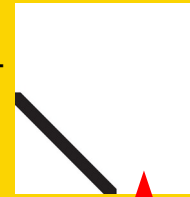


Aufgabe 2:

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.)

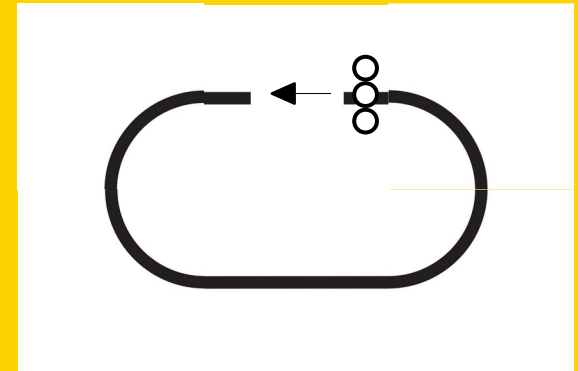
Tauscht eine Kurven-Kachel gegen eine Diagonale aus.



Aufgabe 3:

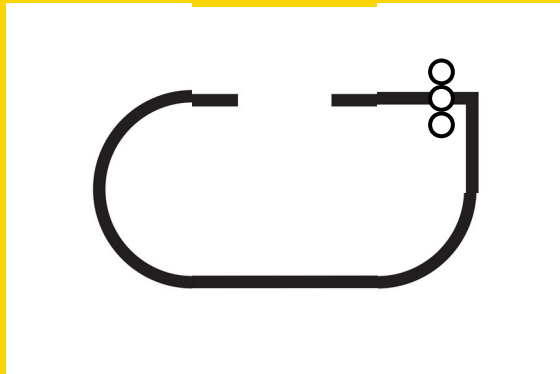
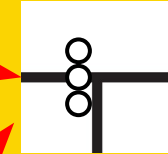
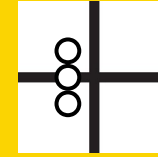
Baut in das Programm den mittleren Sensor mit ein.

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



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

Linker Sensor	Mittlerer Sensor	Rechter Sensor	Situation
0	0	0	(0) Kreuzung
0	0	1	(1) Abknicken nach links (oder Kreuzung?)
0	1	0	(2) Geht das?
0	1	1	(3) Linie verloren
1	0	0	(4) Abknicken nach rechts (oder Kreuzung?)
1	0	1	(5) Mittig auf der Linie
1	1	0	(6) Linie verloren
1	1	1	(7) Gap



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 Millisekunden (ms) vorwärts fahren entspricht welcher Strecke?
(Lasst den Roboter 1 Sekunde (s) 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