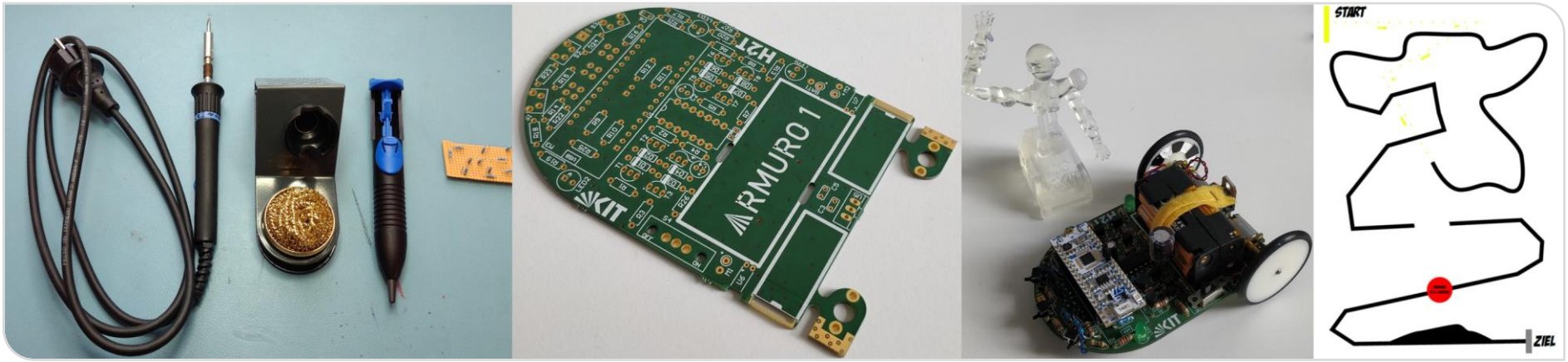
Karlsruher Institut für Technologie

Hochperformante Humanoide Technologien

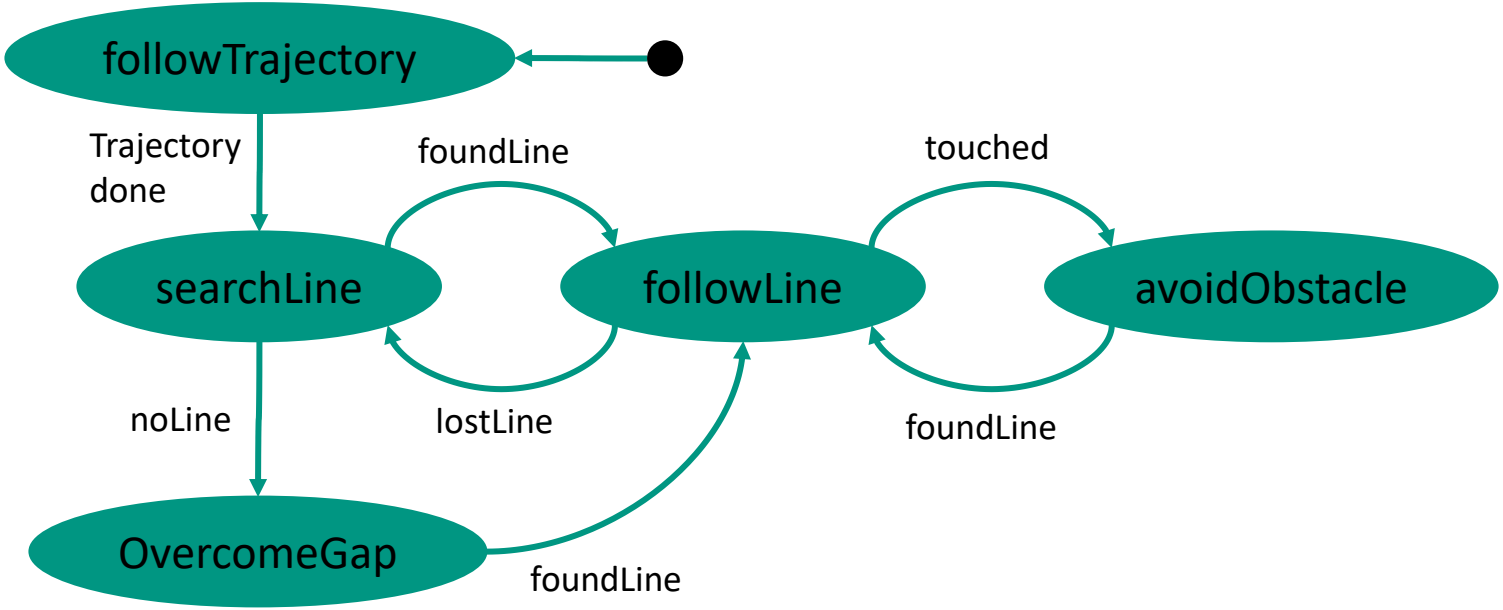
Basispraktikum Mobile Roboter im SoSe 2024

Kolloquium 08 am 26.06.2024



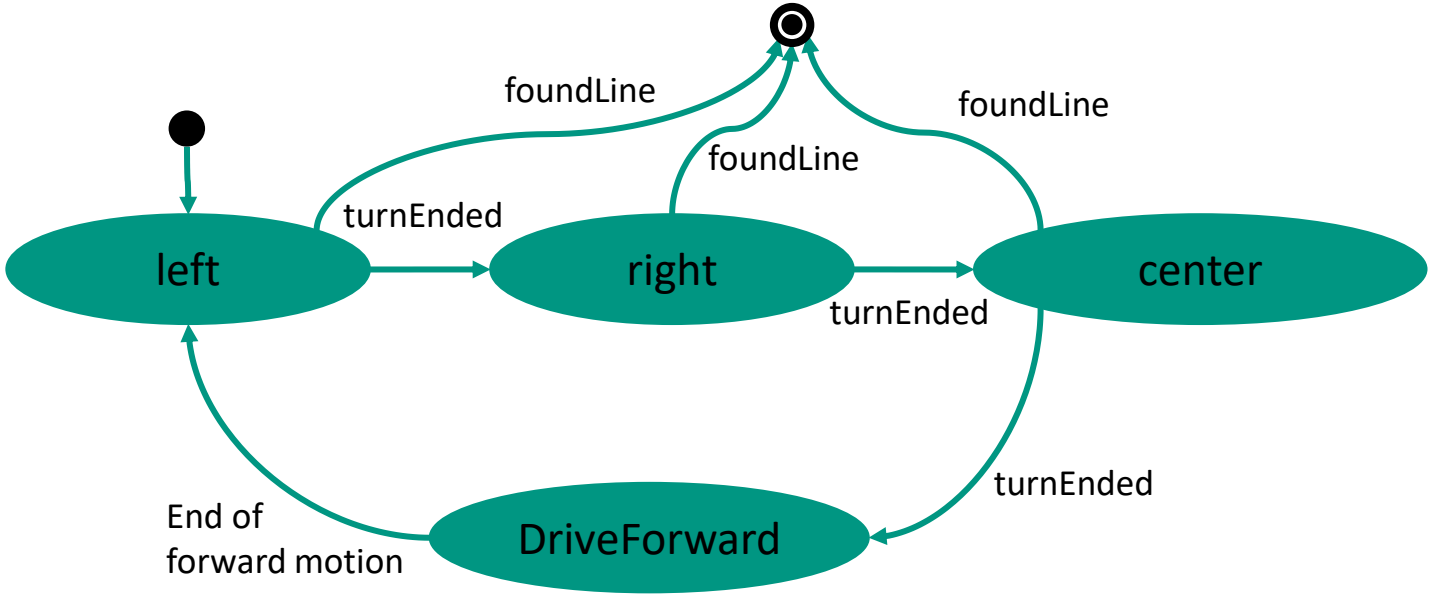
Frage 1.1

Skizzieren sie den Zustandsautomaten mit den sechs oben genannten Zuständen sowie allen sinnvollen Transitionen und beschriften Sie diese.



Frage 2.1

Entwerfen Sie einen Zustandsautomaten, der alle Transitionen und Zustände des zugehörigen searchLine-Tasks enthält.



Frage 2.2

Implementieren Sie den kooperativen Task task_searchLine() gemäß ihrer Überlegungen in 2.1

Simeon Schrape

```
// AUFGABE 2.2
 case SEARCHLINEL:
     switch(search_phase){
         case TURNLEFT:
             if(turn_left_amount(100))search_phase = TURNBACKL;
             if(check_for_line()){
                 state = FOLLOWLINE;
                 phase = INIT;
             }
             break;
         case TURNBACKL:
             if(turn_right_amount(100))search_phase = TURNRIGHT;
             break;
         case TURNRIGHT:
             if(turn_right_amount(100))search_phase = TURNBACKR;
             if(check_for_line()){
                 state = FOLLOWLINE;
                 phase = INIT;
             }
             break;
         case TURNBACKR:
             if(turn_left_amount(110)){
                 state = OVERCOMEGAP;
                 gap_phase = GAP_INIT;
             }
             break;
     }
     break;
 case SEARCHLINER:
     switch(search_phase){
         case TURNRIGHT:
             if(turn_right_amount(100))search_phase = TURNBACKR;
             if(check_for_line()){
                 state = FOLLOWLINE;
                 phase = INIT;
             }
             break;
         case TURNBACKR:
             if(turn_left_amount(100)){
                 search_phase = TURNLEFT;
             }
             break;
         case TURNLEFT:
             if(turn_left_amount(100))search_phase = TURNBACKL;
             if(check_for_line()){
                 state = FOLLOWLINE;
                 phase = INIT;
             }
             break;
         case TURNBACKL:
             if(turn_right_amount(100)){
                 state = OVERCOMEGAP;
                 gap_phase = GAP_INIT;
             }
             break;

     }
     break;
```

Frage 3.1

Implementieren Sie den kooperativen Task task_avoidObstacle() um das Hindernis auf der Linie zu umfahren.

David Sanytskyi

```
// Aufgabe 3 Begin
typedef enum State_AvoidObstacle {
    AvoidObstacle_Turn1 = 0,
    AvoidObstacle_Drive1 = 1,
    AvoidObstacle_Turn2 = 2,
    AvoidObstacle_Drive2 = 3,
    AvoidObstacle_Finish = 4
} State_AvoidObstacle_Enum;
State_AvoidObstacle_Enum State_AvoidObstacle;
void Start_Task_AvoidObstacle() {
    Tasks.AvoidObstacle.status = Enabled;
}
void Task_AvoidObstacle() {
    if (Tasks.Turn.status || Tasks.DriveForward.status)
        return;
    switch (State_AvoidObstacle) {
    case AvoidObstacle_Turn1:
        Start_Task_Turn(Forward, Left, 45);
        break;
    case AvoidObstacle_Drive1:
        Start_Task_DriveForward(75);
        break;
    case AvoidObstacle_Turn2:
        Start_Task_Turn(Forward, Right, 90);
        break;
    case AvoidObstacle_Drive2:
        Start_Task_DriveForward(75);
        break;
    case AvoidObstacle_Finish:
        Tasks.AvoidObstacle.status = Disabled;
    }
    State_AvoidObstacle++;
}
// Aufgabe 3 End
```

Organisatorisches

- **Abschlussrennen**:
  - Datum: 10.07.24 (übernächste Woche) um 14:00 Uhr
  - Wir werden die Parameter für den ersten Teil des Parcours am Termin in der Woche davor bekanntgeben (Anzahl der Winkel und Strecken bleibt im Vergleich zum Test-Parcours unverändert)
  - Es ist ein **Rennen,** der schnellste gewinnt einen Preis
  - Für den Parcours gilt ein Zeitlimit von 150 Sekunden Fahrtzeit
  - Regeln für das Rennen auf der nächsten Folie.

- Der Termin am 03.07. (nächste Woche) ist optional.
  - Ihr habt Gelegenheit euren Code zu verbessern und zu testen

- Teil der Prüfungsleistung ist die Abgabe des Codes inclusive Dokumentation
  - **Deadline** ist der 24.07.24 23:59 Uhr

Regeln Abschlussrennen

- Jeder Teilnehmer hat erstmal einen Versuch
- Falls eingegriffen werden muss, (Roboter verlässt Linie, dreht um, etc.) gibt es 10 Straf-Sekunden. Der Roboter muss wieder vor das Hindernis gesetzt werden
- Muss neu gestartet werden gibt es 15 Straf-Sekunden, man darf neu programmieren und der nächste Versuch findet zum Schluss statt
- Ist die Gesamtzeit größer als 150 Sekunden so ist der Teilnehmer durchgefallen

Karlsruher Institut für Technologie

Fragen?