

Sumoroboti töötoa ettevalmistamine

Enne igat koolitust tuleb veenduda, et:

1. Arvutitel on Arduino ja Sumoroboti teek installeeritud:
 - lae alla ja installeeri Arduino <http://www.arduino.cc/download>
 - lae alla ja installeeri Sumoroboti teek <https://github.com/silps/sumoino>
2. Sumoroboti akud oleks täidetud:
 - multimeetriga mõõda ligikaudu 7,4V aku kohta
3. Sumorobotil töötavad näiteprogrammid:
 - lae Sumorobotisse näiteprogramm



Juhul, kui midagi ei tööta:

Vaata üle kas juhtmed on õigesti ühendatud:

- parem mootor nr 6 ja vasak nr 5 pistikus
- vastase parem andur nr 0 ja vasak nr 1 pistikus
- joone parem andur nr 2, keskmine nr 3 ja vasak nr 4 pistikus

Kasuta teeki sisse ehitatud “test” funktsioone:

- testOpponent(); (nähes vastast, ründab)
- testLine(); (sõidab otse ja väldib joont)
- testSensors(); (väljastab kõik anduri väärtused konsooli)

Kalibreeri võimalike väärtusi “Sumo.h” failis:

- valguse intensiivsust jooneanduril (LIGHT_INTENSITY)
- vastase nägemise kaugust (OPPONENT_DISTANCE)
- servo mootorite keskpunkti, ehk stopp punkti (MIDPOINT)

```
32 /* configuration */
33 /* movement */
34 #define SECOND 1000
35 #define SEKUND 1000
36 #define MAX_SPEED 100 // 100
37 #define SLOW 10 // 10%
38 #define FORWARD 0 // for
39 #define EDASI 0
40 #define BACKWARD 1 // bac
41 #define TAGASI 1
42 #define MAX_FORWARD 180 // max
43 #define MIDPOINT 94 // ser
44 #define MAX_BACKWARD 0 // max
45 /* pins */
46 #define RIGHT_WHEEL 6
47 #define LEFT_WHEEL 5
48 #define FRONT_RIGHT_SENSOR 0
49 #define FRONT_LEFT_SENSOR 1
50 #define BOTTOM_RIGHT_SENSOR 2
51 #define BOTTOM_MIDDLE_SENSOR 3
52 #define BOTTOM_LEFT_SENSOR 4
53 /* calibration */
54 #define LIGHT_INTENSITY 600
55 #define OPPONENT_DISTANCE 200
```

Sumorobot Workshop preparation

Before the workshop, be sure that:

1. The computers have Arduino and the Sumorobot library installed:
 - download and install Arduino <http://www.arduino.cc/download>
 - download and install the Sumorobot library <https://github.com/silps/sumoino>
2. The batteries of the Sumorobots are charged:
 - measure approximately 7,4V per battery
3. The example programs are working on the Sumorobots:
 - upload the examples to the Sumorobots



In case something doesn't work:

Check that all cables are connected correctly:

- right motor nr 6 and left nr 5 pin
- opponent right sensor nr 0 and left nr 1 pin
- line right sensor nr 2, middle nr 3 and left nr 4 pin

Use the built-in “test” functions:

- testOpponent(); (attacks the opponent, when in range)
- testLine(); (drives forward and avoids line)
- testSensors(); (prints out all the sensor values to the console)

Calibrate possible values in “Sumo.h” file:

- the light intensity of the line sensor (LIGHT_INTENSITY)
- the opponent detection distance (OPPONENT_DISTANCE)
- servo motor midpoint or stopping point (MIDPOINT)

```
32 /* configuration */
33 /* movement */
34 #define SECOND 1000
35 #define SEKUND 1000
36 #define MAX_SPEED 100 // 100
37 #define SLOW 10 // 10%
38 #define FORWARD 0 // for
39 #define EDASI 0
40 #define BACKWARD 1 // bac
41 #define TAGASI 1
42 #define MAX_FORWARD 180 // max
43 #define MIDPOINT 94 // ser
44 #define MAX_BACKWARD 0 // max
45 /* pins */
46 #define RIGHT_WHEEL 6
47 #define LEFT_WHEEL 5
48 #define FRONT_RIGHT_SENSOR 0
49 #define FRONT_LEFT_SENSOR 1
50 #define BOTTOM_RIGHT_SENSOR 2
51 #define BOTTOM_MIDDLE_SENSOR 3
52 #define BOTTOM_LEFT_SENSOR 4
53 /* calibration */
54 #define LIGHT_INTENSITY 600
55 #define OPPONENT_DISTANCE 200
```

Sumoroboter Workshop Vorbereitung

Vor dem Workshop, sei sicher dass:

1. Die Computer haben Arduino und die Sumoroboter Bibliothek installiert:
 - downloade und installiere Arduino <http://www.arduino.cc/download>
 - downloade und installiere die Bibliothek <https://github.com/silps/sumoino>
2. Die Batterien von den Sumorobotern sind voll:
 - messe ungefähr 7,4V pro Batterie
3. Die Beispiel Programme funktionieren auf den Sumorobotern:
 - uploade die Beispiele auf die Sumoroboter



Im Falle das was nicht funktioniert:

Siehe ob alle Kabel richtig verbunden sind:

- rechter Motor nr 6 und linker nr 5 pin
- rechter Gegner-Sensor nr 0 und linker nr 1 pin
- rechter Linien-Sensor nr 2, mittlerer nr 3 und linker nr 4 pin

Benutze die built-in “test” Funktionen:

- testOpponent(); (greift den Gegner an, wenn im Umfeld)
- testLine(); (fährt vorwaerts und vermeidet die Linie)
- testSensors(); (schick alle Sensor Daten in die Konsole)

Kalibriere mögliche Werte in File “Sumo.h”:

- die Intensität vom Lichtsensor (LIGHT_INTENSITY)
- die Gegnererkennung-Distanz (OPPONENT_DISTANCE)
- der Mittelwert oder Stoppwert vom Servo (MIDPOINT)

```
32 /* configuration */
33 /* movement */
34 #define SECOND 1000
35 #define SEKUND 1000
36 #define MAX_SPEED 100 // 100
37 #define SLOW 10 // 10%
38 #define FORWARD 0 // for
39 #define EDASI 0
40 #define BACKWARD 1 // bac
41 #define TAGASI 1
42 #define MAX_FORWARD 180 // max
43 #define MIDPOINT 94 // ser
44 #define MAX_BACKWARD 0 // max
45 /* pins */
46 #define RIGHT_WHEEL 6
47 #define LEFT_WHEEL 5
48 #define FRONT_RIGHT_SENSOR 0
49 #define FRONT_LEFT_SENSOR 1
50 #define BOTTOM_RIGHT_SENSOR 2
51 #define BOTTOM_MIDDLE_SENSOR 3
52 #define BOTTOM_LEFT_SENSOR 4
53 /* calibration */
54 #define LIGHT_INTENSITY 600
55 #define OPPONENT_DISTANCE 200
```