

Aufgabenblatt 4 (Praxis)

(10 Punkte)

Hinweis: Abgabe des Theorieteils (einzeln oder in Zweiergruppen) bis zum 08.11.2019 um 10:00 durch Einwurf in den Briefkasten (Erdgeschoss OH16, gegenüber von Raum E16). Eine Abgabe per E-Mail ist *nicht* möglich. Bearbeitung: 11.-15.11.2019.

1 Vorbereitung (3 Punkte)

Hinweis: Diese Aufgabe muss abgegeben werden!

Lesen Sie vor Beginn der praktischen Übung die Kapitel 1 und 4 der OSEK-Specifications und beantworten Sie die folgenden Fragen:

- Wie wird ein Task im OSEK-Betriebssystem terminiert?
- OSEK unterscheidet zwischen zwei Tasktypen. Nennen Sie diese und erläutern Sie den Unterschied.
- In welchem Zustand befindet sich ein vom Scheduler aktivierter Task?

2 EV3OSEK Einrichten (2 Punkte)

Melden Sie sich mit den Zugangsdaten, die Sie in der Übung erhalten haben, an einem Rechner des CI-Labs an. Öffnen Sie den für die aktuelle Übung gedachten Ordner (im Zweifelsfall fragen Sie Ihren Tutor/Ihre Tutorin). Wechseln Sie in den darin befindlichen Ordner `ev3osek`, der wiederum den Ordner `newlib` enthält. Öffnen Sie in `newlib` ein Terminal und kompilieren Sie mit dem Befehl `make` die Standardbibliothek für EV3OSEK.

3 ECRobot API (5 Punkte)

Wechseln Sie in das Verzeichnis `../example/CollisionDetect`, in dem sich die Datei `collision.c` befindet. Öffnen Sie diese mit einem Texteditor.

Ergänzen Sie die durch Kommentare markierten Stellen, sodass der Roboter eine (180-Grad-)Wende vor einem Hindernis macht. Wichtig ist es dabei, zu beachten, dass der Task `CheckDistance` alle 30ms ausgeführt wird. Speichern Sie Ihre Änderungen und kompilieren Sie das Programm mit dem Befehl `make`. Kopieren Sie anschließend die Dateien `boot.scr` und `*.bin` auf die MicroSD-Karte. Wenn die MicroSD-Karte wieder im Roboter platziert wurde, sollte das Programm ausgeführt werden. Sollte das Programm nicht ausgeführt werden, könnte es fehlerhaft sein.

Benutzen Sie zur Lösung dieser Aufgabe bitte die Kapitel 1 und 4 der OSEK-Specifications, die Sie bereits gelesen haben, sowie die folgende API:



Servo motor	Description
<pre>S32 ecrobot_get_motor_rev(U8 port_id)</pre>	<p>Gets Servo Motor revolution value in degree. Wrapper of <code>nxt_motor_get_count</code>.</p> <p>Parameters: <code>port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4</code></p> <p>Returns: Servo Motors revolution in degree</p>
<pre>void ecrobot_set_motor_speed(U8 port_id, S8 speed)</pre>	<p>Sets Servo Motor PWM value. Wrapper of <code>nxt_motor_set_speed</code>, but brake mode is fixed as brake.</p> <p>Parameters: <code>port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4</code> <code>speed: -100 to +100</code></p>
<pre>void ecrobot_set_motor_mode_speed(U8 port_id, S32 mode, S8 speed)</pre>	<p>Sets Servo Motor brake mode and PWM value. Wrapper of <code>nxt_motor_set_speed</code>.</p> <p>Parameters: <code>port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4</code> <code>mode: 0(float), 1(brake)</code> <code>speed: -100 to +100</code></p>
<pre>void ecrobot_set_motor_rev(U8 port_id, S32 rev)</pre>	<p>Sets Servo Motor revolution value in degree. Wrapper of <code>nxt_motor_set_count</code>.</p> <p>Parameters: <code>port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4</code> <code>rev: Servo Motors revolution in degree</code></p>

Abbildung 1: Motors API.



Ultrasonic sensor	Description
<pre>void ecrobot_init_sonar_sensor(U8 port_id)</pre>	<p>Init a NXT sensor port for Ultrasonic Sensor.</p> <p>Parameters: port_id: EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D</p>
<pre>S32 ecrobot_get_sonar_sensor(U8 port_id)</pre>	<p>Get Ultrasonic Sensor measurement data in cm.</p> <p>Parameters: port_id: EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D</p> <p>Returns: Distance in cm (0 to 255), -1 (failure)</p>
<pre>void ecrobot_get_sonar_sensor_single_shot(U8 port_id, U8 data_buffer[8])</pre>	<p>Set the mode of the Lego ultrasonic sensor at the specified port to ULTRASONIC_MODE_SINGLE_SHOT. After that get the range of the Lego ultrasonic sensor connected at the specified port and store it in the buffer. The sensor measures distances from 0 to 255 in cm. If nothing is located in front of the sensor, the value will be 255. All 8 entries of the array will be values returned by the sensor. If less than 8 objects are detected, some entries will be set to 255.</p> <p>Parameters: port_id: EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D data_buffer: Buffer to store the result in</p>
<pre>void ecrobot_term_sonar_sensor(U8 port_id)</pre>	<p>Terminate I2C used for for Ultrasonic.</p> <p>Parameters: EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D</p>

Abbildung 2: Ultrasonic Sensor API.

Allgemeine Hinweise: Alle Übungstermine und weitere Informationen zur Veröffentlichung und Abgabe der Übungszettel sowie zum Erreichen der Studienleistung finden Sie unter

<https://1s12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/wintersemester-2019/es-1819.html.html>.