

## Exercise Sheet 4 (Practice)

(11 Points)

**Please note:** Solutions must be submitted (individually or in pairs) until 08.11.2019 at 10:00 AM (mailbox in OH16, ground floor, in front of room E16). Submitting solutions via mail is *not* possible. Discussion: 11.-15.11.2019.

### 1 Preparation (3 Points)

**Please note: The solution to this assignment must be handed in!**

Previous to the exercise session, read chapters 1 and 4 of the OSEK specifications. Answer the following questions:

- How are tasks terminated in the OSEK operating system?
- In OSEK, two types of tasks can be distinguished. Name both types and explain the difference.
- In which state is a task that was activated by the scheduler?

### 2 EV3OSEK Setup (3 Points)

Use the credentials you received at the beginning of the exercise session for logging in. Down In the CI-Lab, log in using the credentials you received. Open the folder containing the material for the current exercise session (if in doubt, ask your tutor). Change to the folder `ev3osek`, in which a folder `newlib` is located. In `newlib`, open a terminal and enter `make` to compile the EV3OSEK standard library. Thereon, proceed to the next assignment.

### 3 ECRobot API (5 Points)

Change into the directory `../example/CollisionDetect` which contains the file `collision.c`. Open the file with your text editor of choice.

Complete the code as indicated in the comments, such that the robot performs a (U-)turn when detecting an obstacle. Please take into consideration, that the task `CheckDistance` is executed every 30ms. Save your changes and compile the program with the command `make`. Thereon, copy the files `boot.scr` and `*.bin` to the MicroSD card. When the card is placed in the robot, the program should be executed. If it is not executed, your program may contain errors.

To solve this assignment, please use chapters 1 and 4 of the OSEK specifications, which you already read in advance, as well as the following 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.

**General information:** Further information about the exercises, exercise sheets, and the exam admission can be found at

<https://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/wintersemester-2019/es-1819.html.html>.