lea.schoenberger [☺] tu-dortmund.de
nils.hoelscher [☺] tu-dortmund.de
nick.pietrass [☺] tu-dortmund.de
jan.pomplun [☺] tu-dortmund.de

Exercises for
Embedded Systems
Wintersemester 18/19

Exercise Sheet 4 (Practice)

(10 Points)

Please note: Solutions must be submitted (individually or in pairs) until 12.11.2018 at 10:00 AM (post box in OH16, basement, in front of room E16). Submitting solutions via mail is *not* possible. Discussion: 14.-16.11.2018.

1 Preparation (3 Points)

Please note: The solution to this assignment must be submitted!

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? Which special characteristic does it have?

2 EV3OSEK Setup (2 Points)

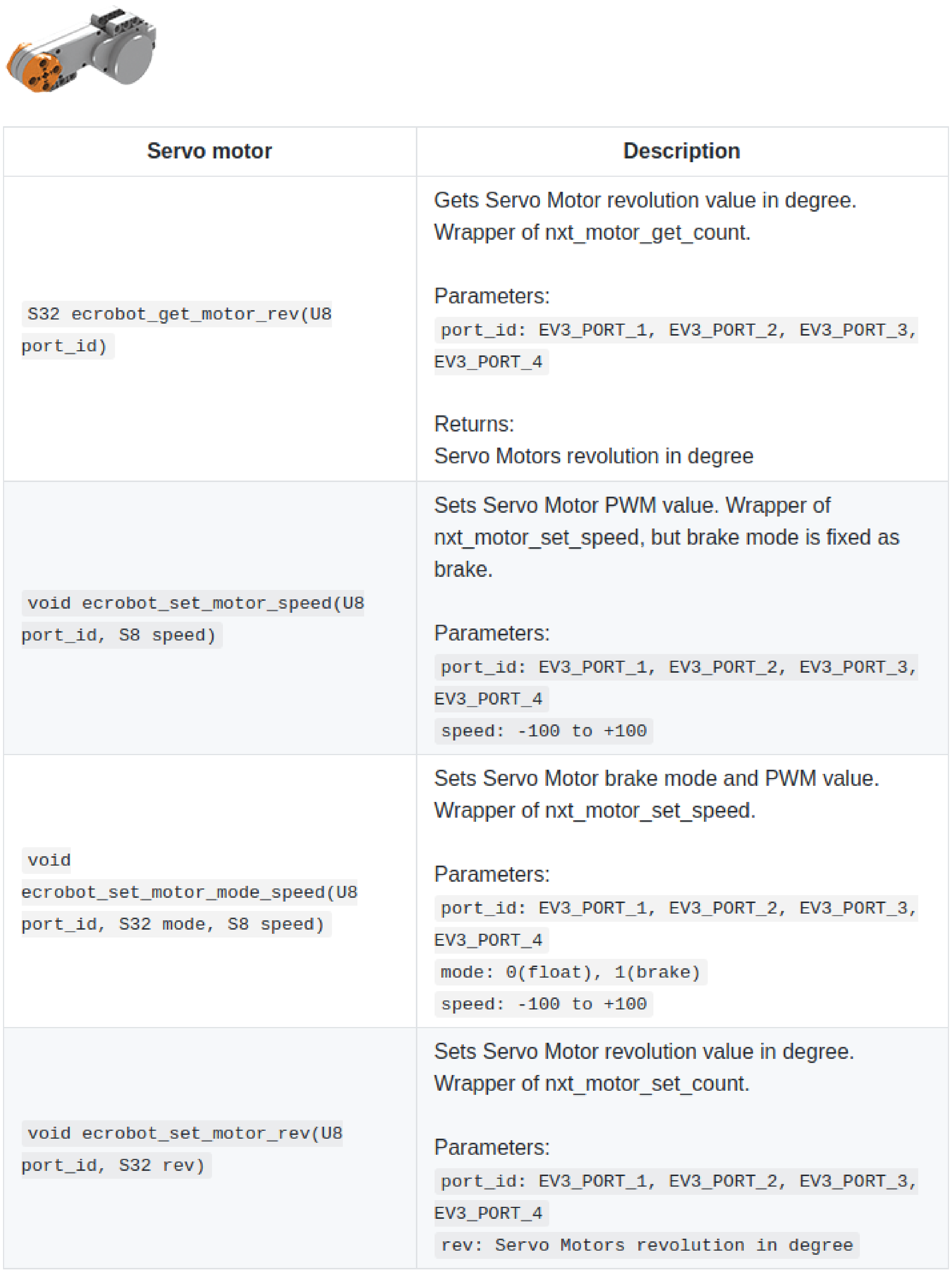
In the CI-Lab, choose the virtual machine `es` and log on. Under `media/nfs/es`, the folders `ev3osek` and `04` are located. Copy the folder `ev3osek` to your home directory and the content of the folder `04` into the folder `example`, which is located in `ev3osek`. Change into the directory `newlib` in `ev3osek`, 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.

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:



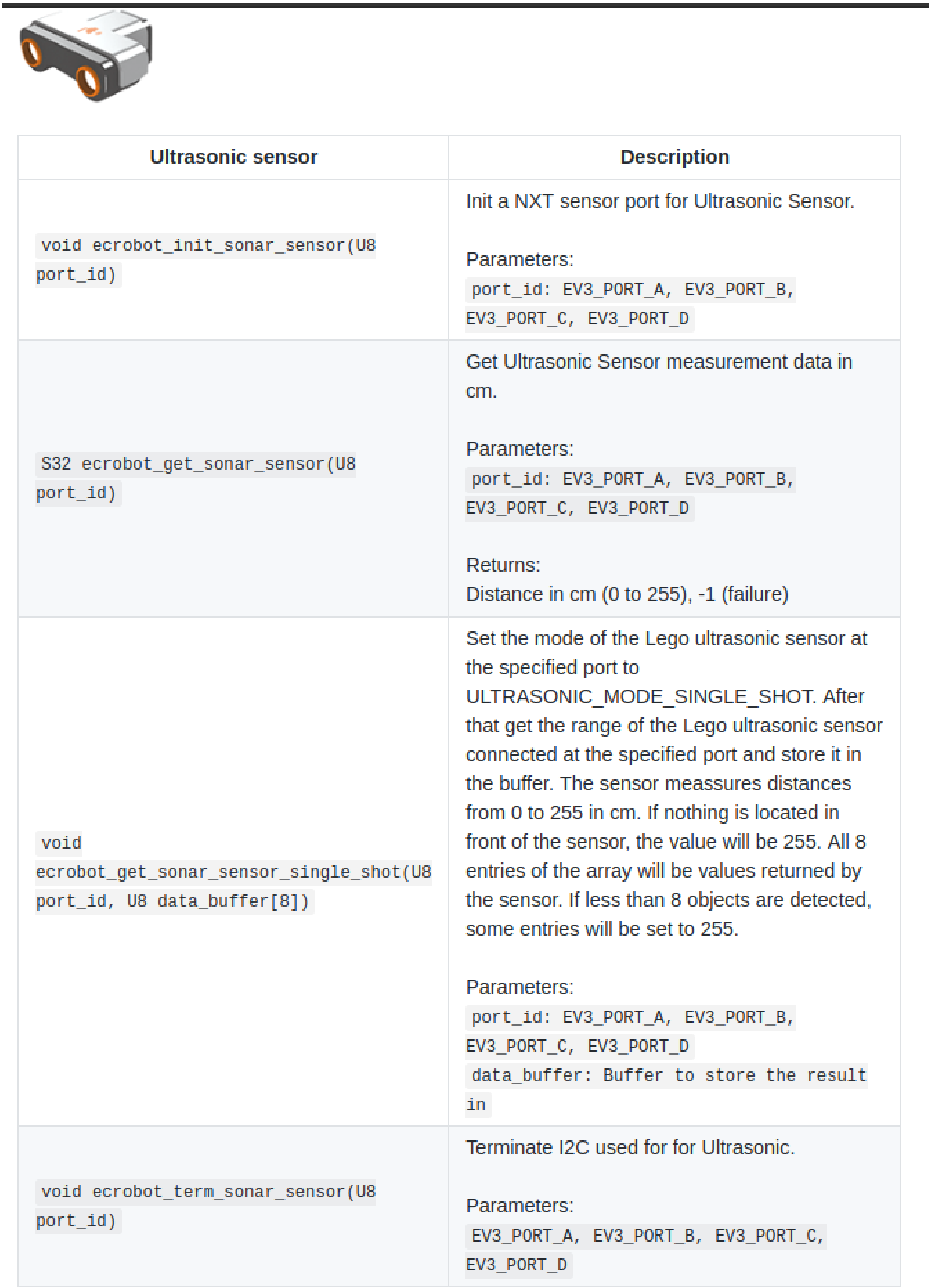
Servo motor	Description
`S32 ecrobot_get_motor_rev(U8 port_id)`	Gets Servo Motor revolution value in degree. Wrapper of nxt_motor_get_count. Parameters: `port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4` Returns: Servo Motors revolution in degree

`void ecrobot_set_motor_speed(U8 port_id, S8 speed)`	Sets Servo Motor PWM value. Wrapper of nxt_motor_set_speed, but brake mode is fixed as brake.
	Parameters:
	`port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4`
	`speed: -100 to +100`

`void ecrobot_set_motor_mode_speed(U8 port_id, S32 mode, S8 speed)`	Sets Servo Motor brake mode and PWM value. Wrapper of nxt_motor_set_speed.
	Parameters:
	`port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4`
	`mode: 0(float), 1(brake)`
	`speed: -100 to +100`

`void ecrobot_set_motor_rev(U8 port_id, S32 rev)`	Sets Servo Motor revolution value in degree. Wrapper of nxt_motor_set_count.
	Parameters:
	`port_id: EV3_PORT_1, EV3_PORT_2, EV3_PORT_3, EV3_PORT_4`
	`rev: Servo Motors revolution in degree`

Abbildung 1: Motors API.



Ultrasonic sensor	Description
`void ecrobot_init_sonar_sensor(U8 port_id)`	Init a NXT sensor port for Ultrasonic Sensor. Parameters: `port_id: EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D`
`S32 ecrobot_get_sonar_sensor(U8 port_id)`	Get Ultrasonic Sensor measurement data in cm. Parameters: `port_id: EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D` Returns: Distance in cm (0 to 255), -1 (failure)

`void ecrobot_get_sonar_sensor_single_shot(U8 port_id, U8 data_buffer[8])`	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 meassures 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.
	Parameters:
	`port_id: EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D`
	`data_buffer: Buffer to store the result in`

`void ecrobot_term_sonar_sensor(U8 port_id)`	Terminate I2C used for for Ultrasonic. Parameters: `EV3_PORT_A, EV3_PORT_B, EV3_PORT_C, EV3_PORT_D`

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-2018/es-1819.html`.