Fachhochschule Dortmund
University of Applied Sciences and Arts

GROUP 4

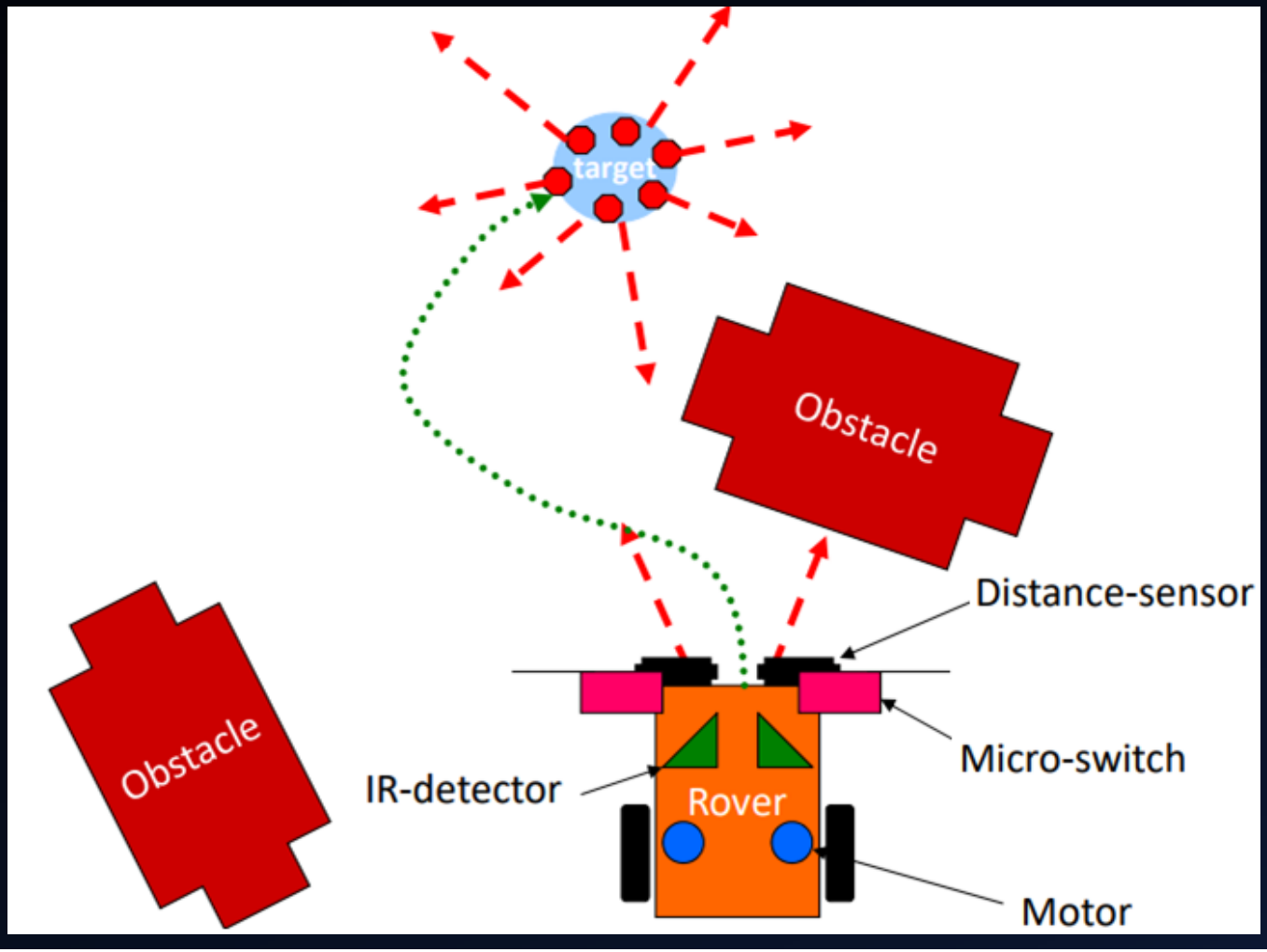
Embedded Systems Lab Project

Cristian Gil (7101834)
Fidel Morles (7104464)

Fachhochschule Dortmund
University of Applied Sciences and Arts

Index

1. Introduction



2. FreeRTOS

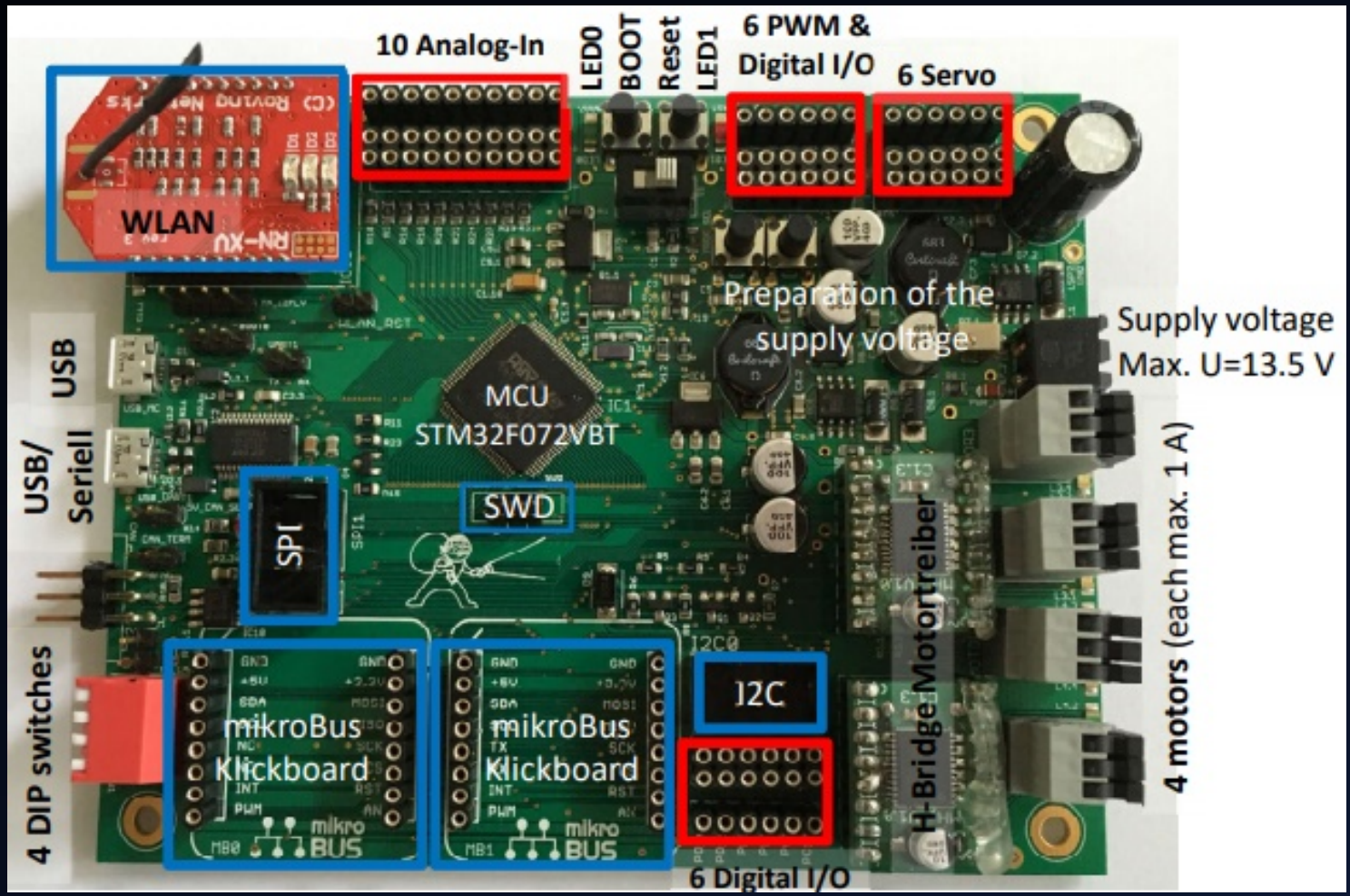
FreeRTOS is a RTOS designed to be small enough to run on a microcontroller – although it is not limited to this application.

Includes:

- Open source RTOS.
- A multitasking scheduler.
- Task priorities system are supported.
- Loads of pre-configured example projects for all supported ports.

Designed to be small and simple

3. DoRoBo32 board

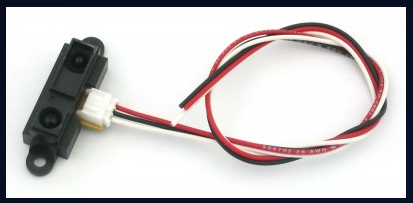


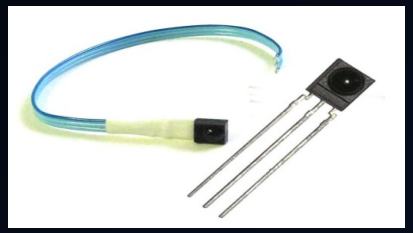
4. Sensors and actuators (I/O)

Fachhochschule Dortmund
University of Applied Sciences and Arts

2x Micro switches
- Digital inputs
- Detection of collisions

2x IR distance sensors
- Analog inputs
- Distance against obstacles



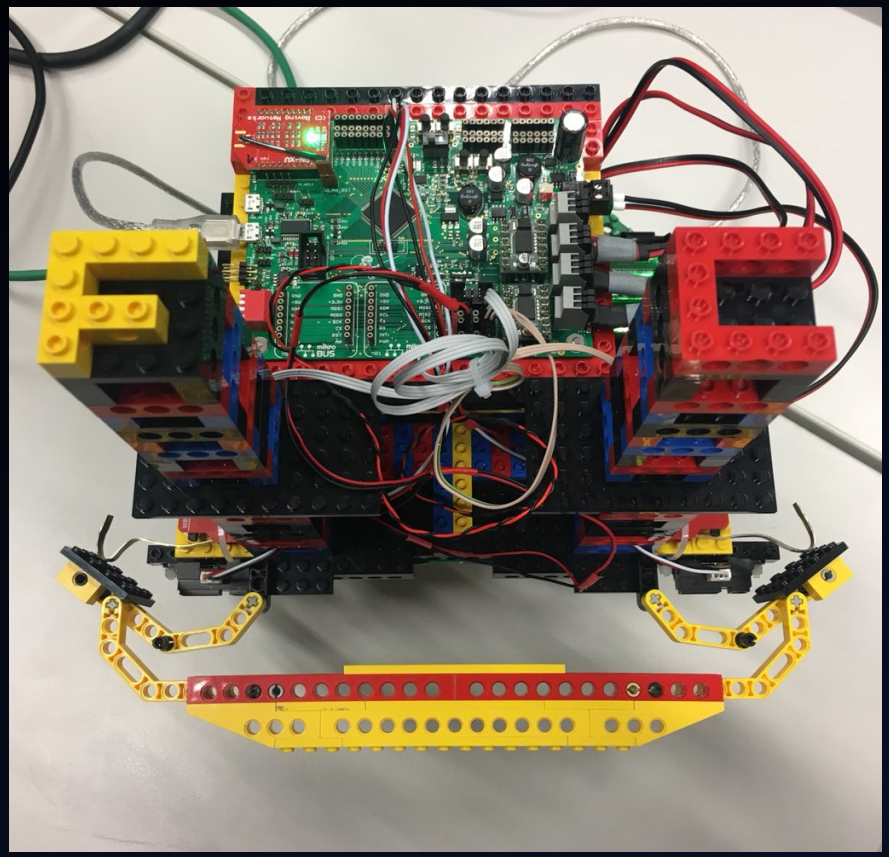


2x IR remote control sensors
- Digital inputs
- Detect a IR transmitter

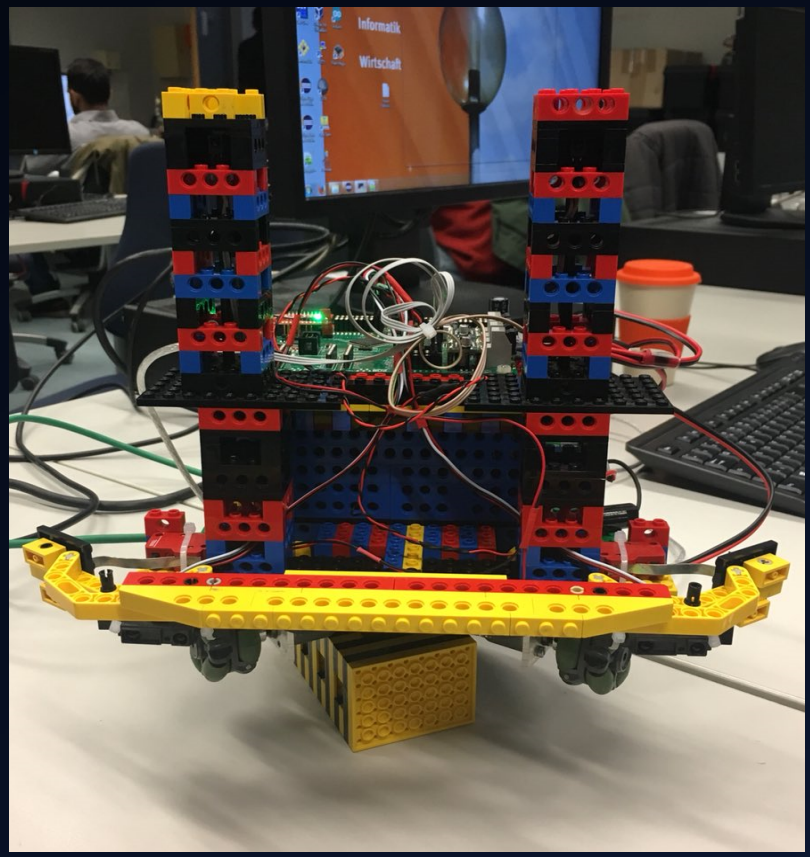
3x Drive motors
- Digital outputs
- Moving the robot

Fachhochschule Dortmund
University of Applied Sciences and Arts

5. Robot design – General view

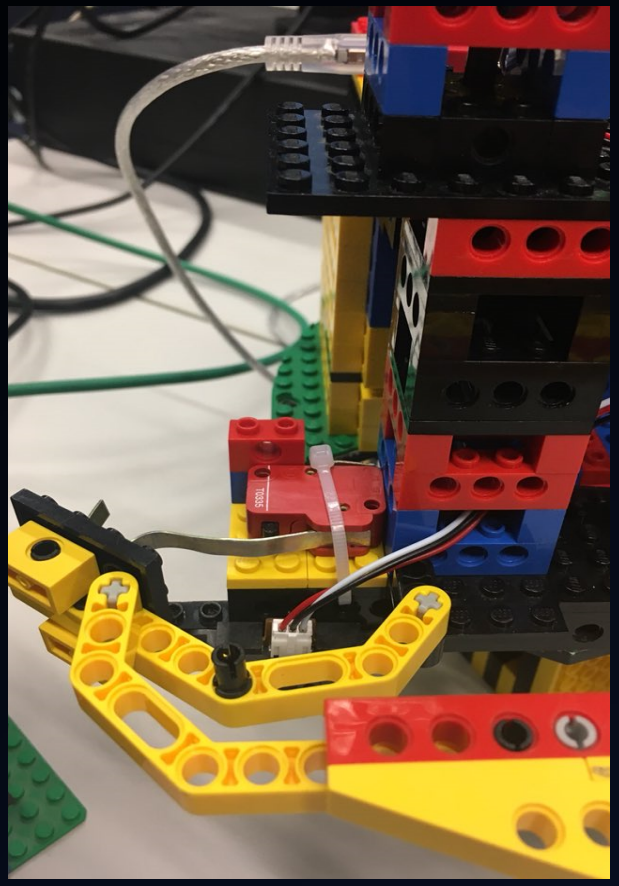


Aerial view

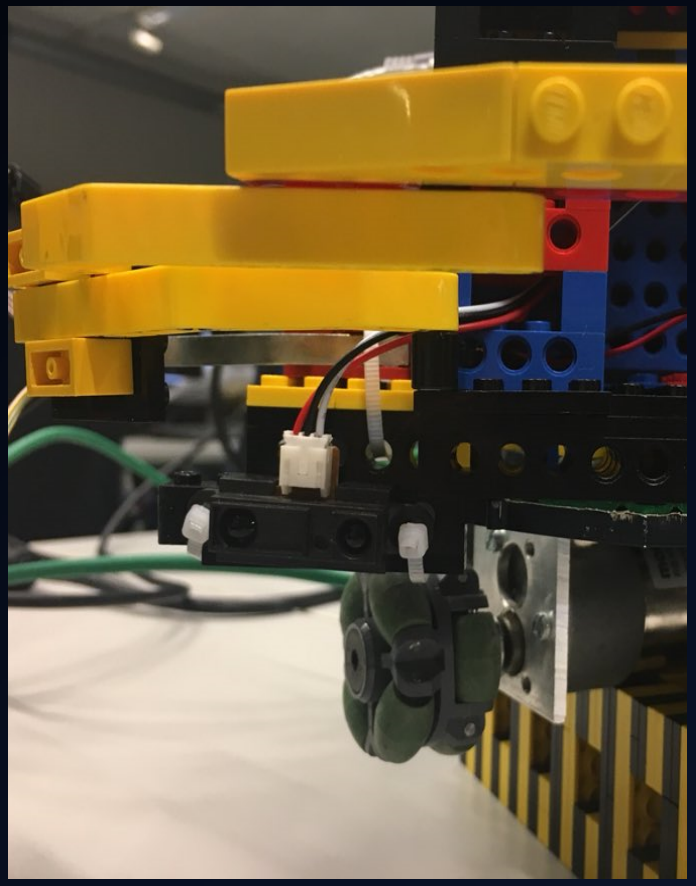


Front view

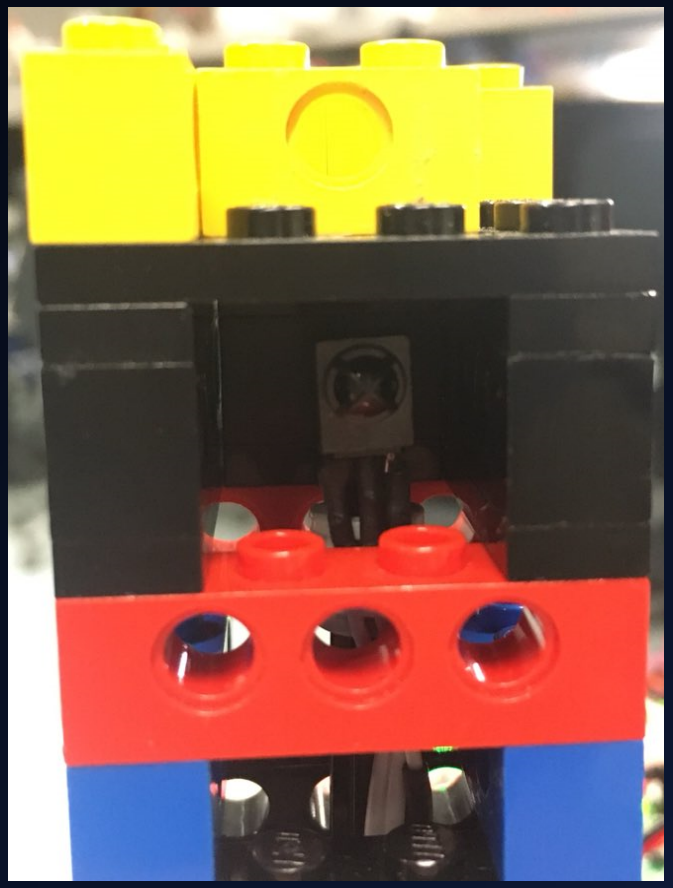
5. Robot design – Detailed view



Micro switch

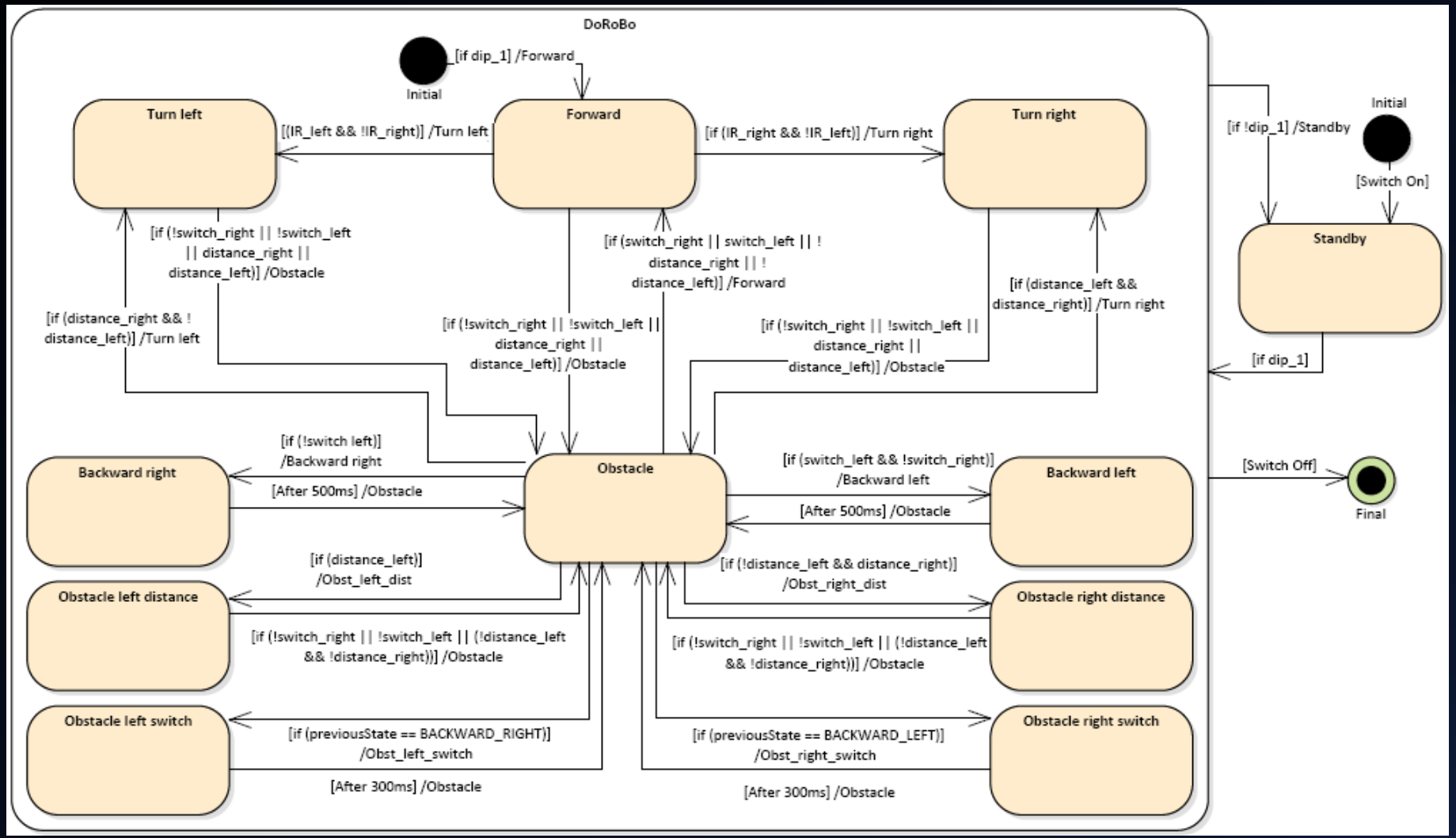


IR distance sensor



IR remote control sensor

6. State machine diagram



7. Source code – Parallel tasks

Fachhochschule Dortmund
University of Applied Sciences and Arts

DoRoBo32, inputs and outputs initializations

....

Inputs configuration

....

```
xTaskCreate (inputs, "INPUTS", 128, NULL, 1, NULL);
xTaskCreate (state_machine, "STATES", 256, NULL, 1, NULL);
xTaskCreate (outputs, "OUTPUTS", 128, NULL, 1, NULL);
```

....

```
vTaskStartScheduler();
```

7. Source code – State machine algorithm

```
if !dip_1 --> STAND_BY

STAND_BY                if dip_1 --> FORWARD
OBSTACLE
  if (!switch_left) --> BACKWARD_RIGHT
  else if (switch_left && !switch_right) --> BACKWARD LEFT
  else if (distance_left) --> OBST_LEFT_DIST
  else if (!distance_left && distance_right) --> OBST_RIGHT_DIST
  else if (previousState == BACKWARD_RIGHT) --> OBST_LEFT_SWITCH
  else if (previousState == BACKWARD_LEFT) --> OBST_RIGHT_SWITCH
  else --> FORWARD
OBST_RIGHT_SWITCH       After 300ms --> OBSTACLE
OBST_LEFT_SWITCH        After 300ms --> OBSTACLE
OBST_RIGHT_DIST
  if (!switch_right || !switch_left || (!distance_left && !distance_right)) --> OBSTACLE
OBST_LEFT_DIST
  if (!switch_right || !switch_left || (!distance_left && !distance_right)) --> OBSTACLE
FORWARD
  if (!switch_right || !switch_left || distance_left || distance_right) --> OBSTACLE
  else if (IR_detection == 1) --> TURN_RIGHT
  else if (IR_detection == -1) --> TURN_LEFT
BACKWARD_RIGHT          After 500ms --> OBSTACLE
BACKWARD_LEFT           After 500ms --> OBSTACLE
TURN_RIGHT
  if (!switch_right || !switch_left || distance_left || distance_right) --> OBSTACLE
  else if (IR_detection == 0) --> FORWARD
TURN_LEFT
  if (!switch_right || !switch_left || distance_left || distance_right) --> OBSTACLE
  else if (IR_detection == 0) --> FORWARD
```

8. Conclusions

- RTOS are **mandatory** in systems, where meeting **deadlines** are important.
- Thought for working with low resources, **RTOS** will be (almost) always the best option for an **embedded system**.
- The current evolution curve of the electronics technology will eventually lead to a **huge popularity** of the RTOS.

Fachhochschule Dortmund
University of Applied Sciences and Arts

THANK YOU FOR YOUR ATTENTION

Any question?