DUCKIETOWN

QuackSquad

Midterm Presentation

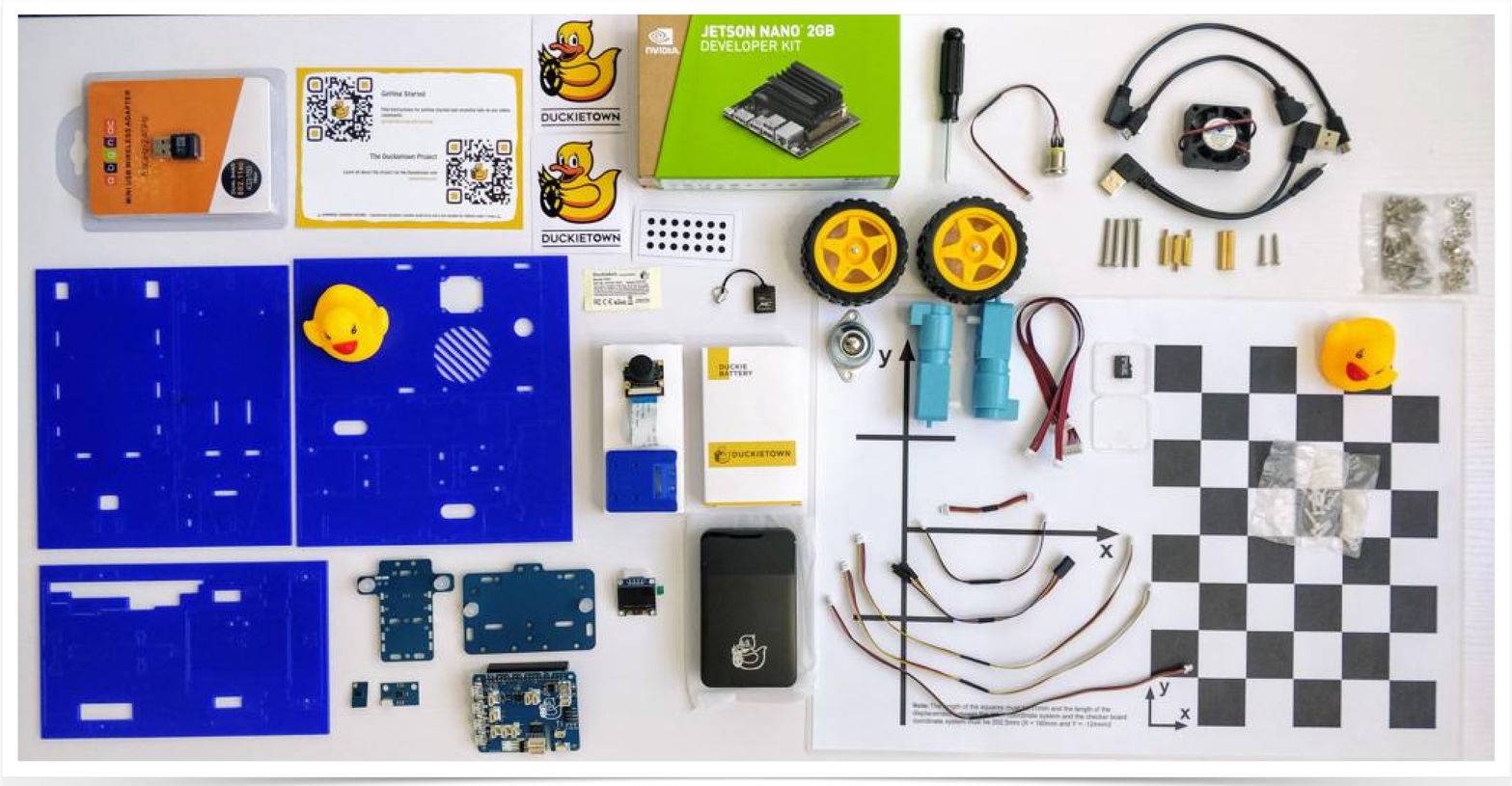
Tuesday, 10 December 2024 | Julien-Alexandre Bertin Klein, Fathia Ismail, Andrea Pellegrin

Step 1

Build

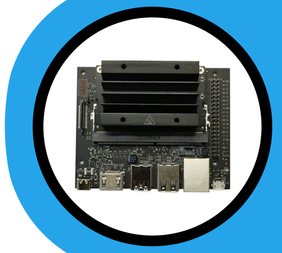
#1 Build

The components



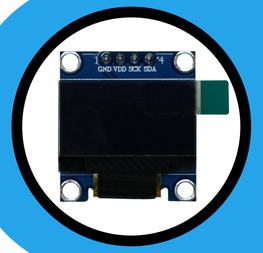
#1 Build

The components

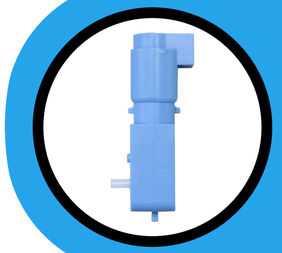


NVIDIA
Jetson Nano

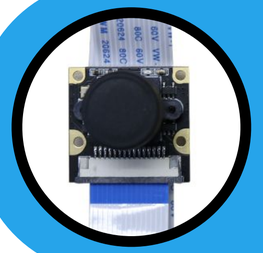
Battery
with diagnostics sensor



Display
128x32 pixels I2C

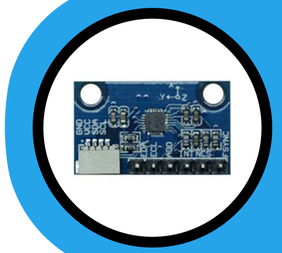


DC motor
with wheel encoder

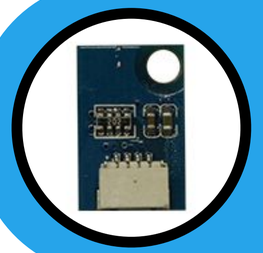


Camera
8 Megapixels

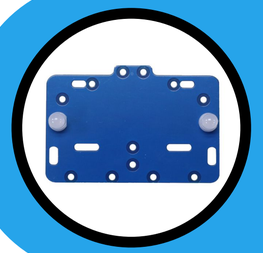
Wi-Fi Adapter



IMU
with gyroscope, accelerometer and magnetometer



Time of Flight



RGB LEDs

#1 Build

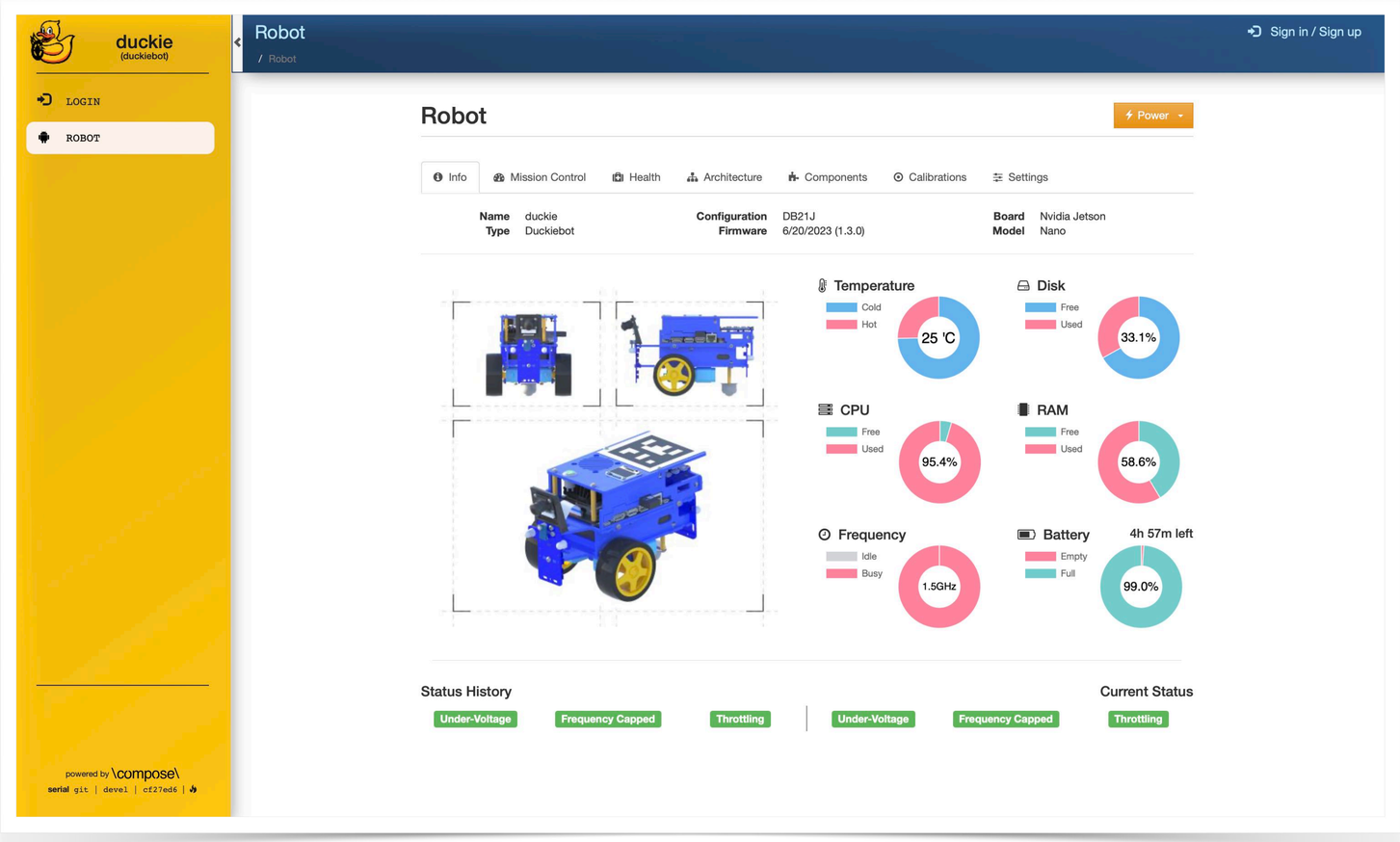
The duckiebot

Step 2

Interfacing

#2 Interfacing

The dashboard



#2 Interfacing

Access via SSH

andreapellegrin — duckie@duckie: ~ — ssh duckie@duckie.local — 80×24

```
andreapellegrin@MacBook-Pro-di-Andrea ~ % ssh duckie@duckie.local
duckie@duckie.local's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.9.140-tegra aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Infrastructure is not enabled.

14 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

194 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

Last login: Mon Dec  9 08:34:01 2024 from 192.168.35.26
duckie@duckie:~$
```

#2 Interfacing

The docker containers

andreapellegrin — duckie@duckie: ~ — ssh duckie@duckie.local — 174×15

```
duckie@duckie:~$ docker ps
CONTAINER ID   IMAGE                                          COMMAND                  CREATED        STATUS                          PORTS     NAMES
1c7755590242   duckietown/dt-core:daffy-arm64v8               "/entrypoint.sh rosl…"   2 minutes ago  Up 2 minutes (healthy)                    demo_lane_following
ca7bf77e3aaa   duckietown/dt-ros-commons:v4.3.0-arm64v8       "/entrypoint.sh bash…"   22 hours ago   Restarting (0) 32 seconds ago             ros
e1e22f9c1c68   duckietown/dt-code-api:v4.1.0-arm64v8          "/entrypoint.sh bash…"   2 weeks ago    Up 3 minutes (healthy)                    code-api
c8d6d56fbb9c   duckietown/dt-duckiebot-interface:v4.3.1-arm64v8 "/entrypoint.sh bash…"  2 weeks ago    Up 3 minutes (healthy)                    duckiebot-interface
772277442c0b   duckietown/dt-device-online:v4.3.0-arm64v8     "/entrypoint.sh bash…"   2 weeks ago    Up 3 minutes (healthy)                    device-online
3f0136a17d5a   duckietown/dt-car-interface:v4.1.0-arm64v8     "/entrypoint.sh bash…"   2 weeks ago    Up 3 minutes (healthy)                    car-interface
41f3d059213b   duckietown/dt-device-health:v4.2.1-arm64v8     "/entrypoint.sh bash…"   2 weeks ago    Up 3 minutes (healthy)                    device-health
9ef3bf2d572a   duckietown/dt-device-proxy:v4.2.0-arm64v8      "/entrypoint.sh bash…"   2 weeks ago    Up 3 minutes (healthy)                    device-proxy
f66796d59980   duckietown/dt-rosbridge-websocket:v4.1.0-arm64v8 "/entrypoint.sh bash…"  2 weeks ago    Up 3 minutes (healthy)                    rosbridge-websocket
f79ff4549759   duckietown/dt-files-api:v4.1.0-arm64v8         "/entrypoint.sh bash…"   2 weeks ago    Up 3 minutes (healthy)                    files-api
cf9620900ae9   duckietown/dt-device-dashboard:v4.1.0-arm64v8  "/entrypoint.sh bash…"   2 weeks ago    Up 3 minutes (unhealthy)                  dashboard
3cf708a6b241   duckietown/portainer:daffy-arm64v8             "/portainer --host=u…"   2 weeks ago    Up 3 minutes                              portainer
duckie@duckie:~$
```

#2 Interfacing

The topics

andreapellegrin — root@duckie: /code/catkin_ws/src/dt-ros-commons — ssh duckie...

```
root@duckie:/code/catkin_ws/src/dt-ros-commons# rostopic list
/diagnostics
/duckie/anti_instagram_node/thresholds
/duckie/auto_calibration_calculation_node/car_cmd
/duckie/auto_calibration_node/car_cmd
/duckie/button_driver_node/event
/duckie/camera_node/camera_info
/duckie/camera_node/image/compressed
/duckie/car_cmd_switch_node/cmd
/duckie/client_count
/duckie/connected_clients
/duckie/coordinator_node/car_cmd
/duckie/diagnostics/code/profiling
/duckie/diagnostics/ros/links
/duckie/diagnostics/ros/node
/duckie/diagnostics/ros/parameters
/duckie/diagnostics/ros/topics
/duckie/display_driver_node/fragments
/duckie/duckiebot_il_lane_following/car_cmd
/duckie/front_center_tof_driver_node/range
/duckie/fsm_node/mode
/duckie/ground_projection_node/debug/ground_projection_image/compressed
/duckie/ground_projection_node/lineseglist_out
/duckie/imu_node/data
/duckie/imu_node/temperature
/duckie/joint_states
/duckie/joy
/duckie/joy/set_feedback
/duckie/joy_mapper_node/car_cmd
/duckie/joy_mapper_node/joystick_override
/duckie/kinematics_node/velocity
/duckie/lane_controller_node/car_cmd
/duckie/lane_controller_node/intersection_navigation_pose
/duckie/lane_filter_node/belief_img
/duckie/lane_filter_node/change_params
/duckie/lane_filter_node/lane_pose
/duckie/lane_filter_node/seglist_filtered
/duckie/lane_pose_visualizer_node/lane_pose_markers
/duckie/lane_recovery_node/car_cmd
/duckie/lane_supervisor_node/car_cmd
/duckie/led_emitter_node/led_pattern
/duckie/left_wheel_encoder_node/tick
/duckie/line_detector_node/debug/edges/compressed
/duckie/line_detector_node/debug/maps/compressed
/duckie/line_detector_node/debug/ranges_HS
/duckie/line_detector_node/debug/ranges_HV
/duckie/line_detector_node/debug/ranges_SV
/duckie/line_detector_node/debug/segments/compressed
/duckie/line_detector_node/segment_list
/duckie/line_segment_visualizer_node/filtered_segment_list_markers
/duckie/line_segment_visualizer_node/segment_list_markers
/duckie/right_wheel_encoder_node/tick
/duckie/road_anomaly_watcher/obstacle_distance
/duckie/simple_stop_controller_node/car_cmd
/duckie/stop_line_filter_node/stop_line_reading
/duckie/velocity_to_pose_node/pose
/duckie/wheels_driver_node/emergency_stop
/duckie/wheels_driver_node/wheels_cmd
/duckie/wheels_driver_node/wheels_cmd_executed
/rosout
/rosout_agg
/tf
/tf_static
root@duckie:/code/catkin_ws/src/dt-ros-commons#
```

#2 Interfacing

Publish to a topic

andreapellegrin — root@duckie: /code/catkin_ws/src/dt-ros-commons — ssh duckie@duckie.local — 160×9

```
root@duckie:/code/catkin_ws/src/dt-ros-commons# rostopic pub /dick/wheels_drive_node/wheels_cmd duckietown_msgs/WheelsCmdStamped -r 1 -- \
> "{header: {seq: 0, stamp: now, frame_id: 'base_link'}, vel_left: 1.0, vel_right: 1.0}"
```

#2 Interfacing

A publisher example

duckietown v3 Current File

dumb_pilot_node.py

```python
class PilotNode(DTROS):  1 usage  Andrea Pellegrin *

    def __init__(self, node_name):  Andrea Pellegrin
        super().__init__(node_name=node_name, node_type=NodeType.CONTROL)

        vehicle_name = os.environ['VEHICLE_NAME']
        topic = f"/{vehicle_name}/wheels_driver_node/wheels_cmd"
        self._publisher = rospy.Publisher(topic, WheelsCmdStamped, queue_size=1)

        self.trim = rospy.get_param(f"/{vehicle_name}/kinematics_node/trim")

    def goStraight(self):  1 usage  Andrea Pellegrin
        vel = 0.5
        calibratedVelLeft = vel * (1 - self.trim)
        calibratedVelRight = vel * (1 + self.trim)
        message = WheelsCmdStamped(vel_left=calibratedVelLeft, vel_right=calibratedVelRight)
        self._publisher.publish(message)
        rospy.sleep(1)

    def stay(self):  1 usage  Andrea Pellegrin
        stop = WheelsCmdStamped(vel_left=0.0, vel_right=0.0)
        self._publisher.publish(stop)
        rospy.sleep(2)

    def run(self):  1 usage  Andrea Pellegrin
        while not rospy.is_shutdown():
            self.goStraight()
            self.stay()

            rospy.sleep(10)

    def on_shutdown(self):  Andrea Pellegrin
        stop = WheelsCmdStamped(vel_left=0.0, vel_right=0.0)
        self._publisher.publish(stop)
```

duckietown > packages > test_package > src > dumb_pilot_node.py 8:24 LF UTF-8 4 spaces Python 3.13

#2 Interfacing

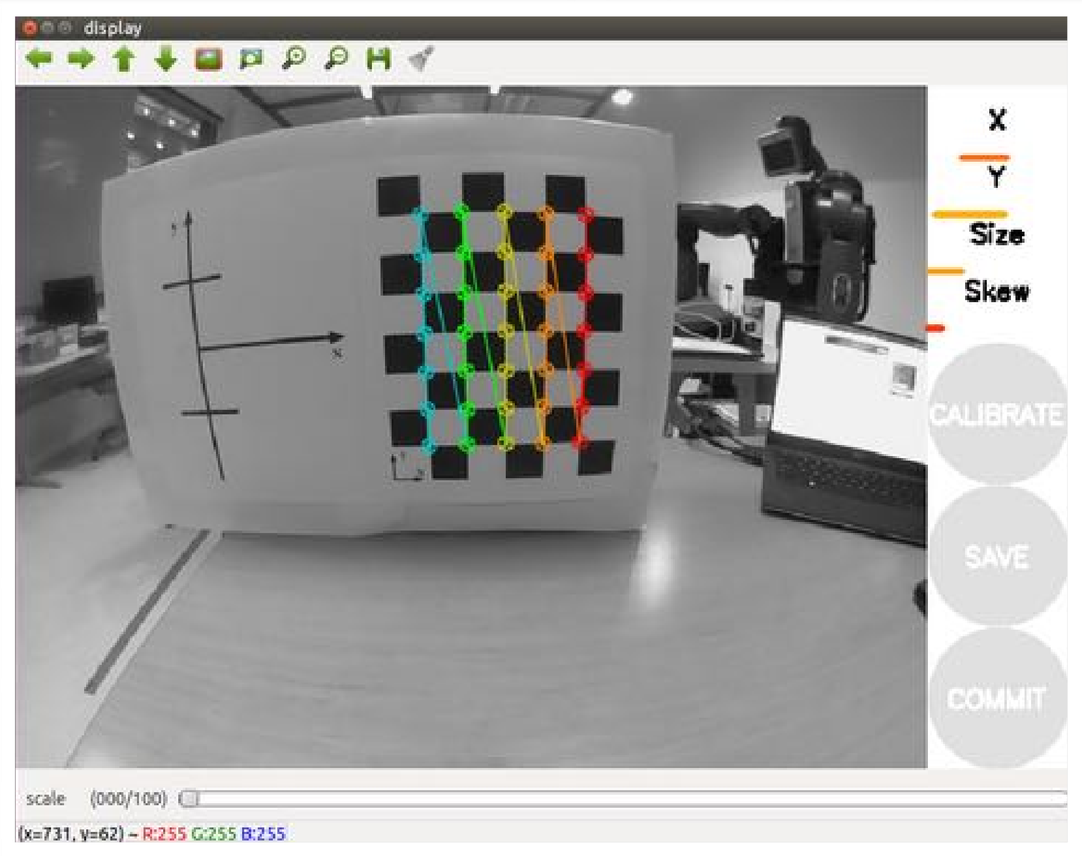
Reading AprilTags

Step 3

Calibration

#3 Calibration

The DuckieTown calibration

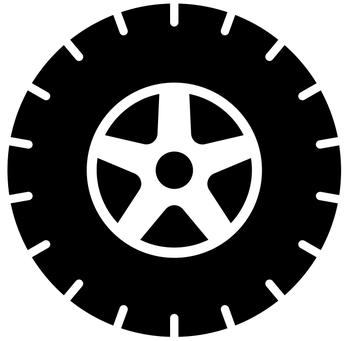


#3 Calibration

The ROS parameters and the use of the encoder

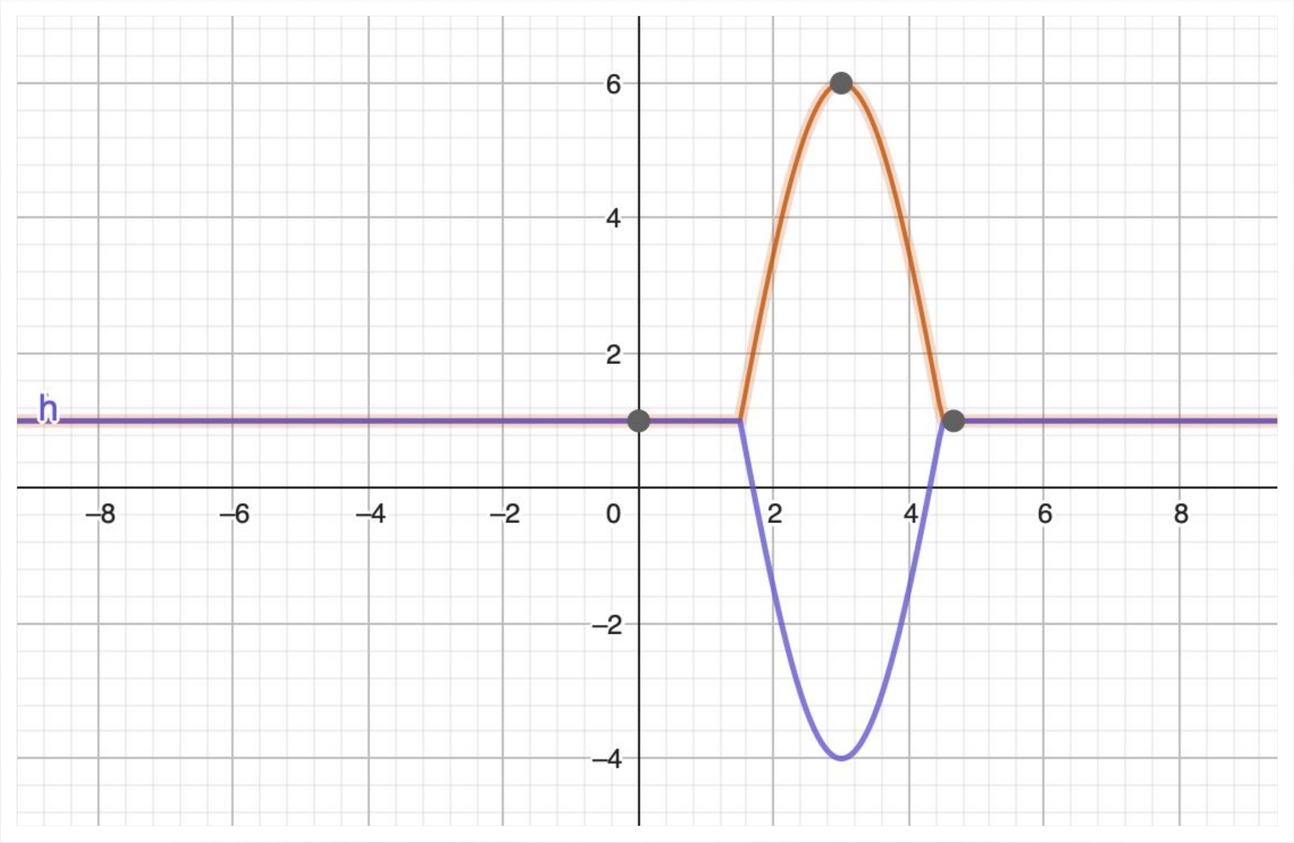
andreapellegrin — duckie@duckie: ~ — ssh duckie@duckie.local — 80×5

```
duckie@duckie:~$ rosparam set /duckie/kinematics_node/trim -0.1
```



#3 Calibration

How to turn?



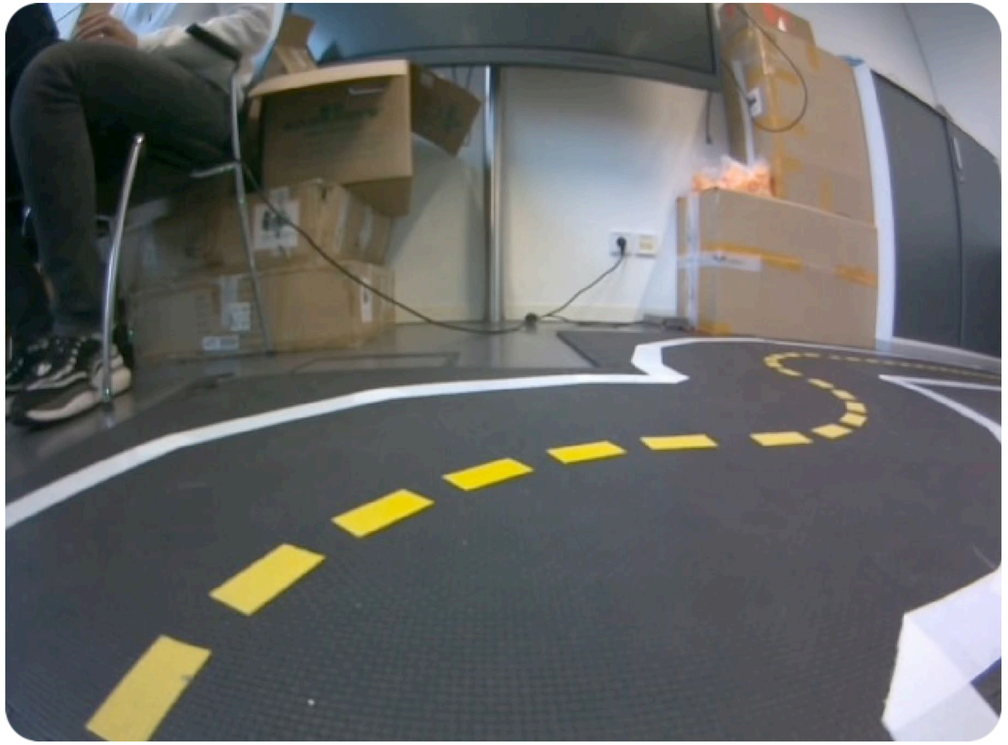
Demo

Let’s dive into!

Demo

Moving around the city

Using the keypad



Using lane following

Future

What’s next?

QuackSquad

Thank you!

Tuesday, 10 December 2024

DUCKIETOWN

QuackSquad

Presentation

Tuesday, 18 March 2025 | Julien-Alexandre Bertin Klein, Fathia Ismail, Andrea Pellegrin

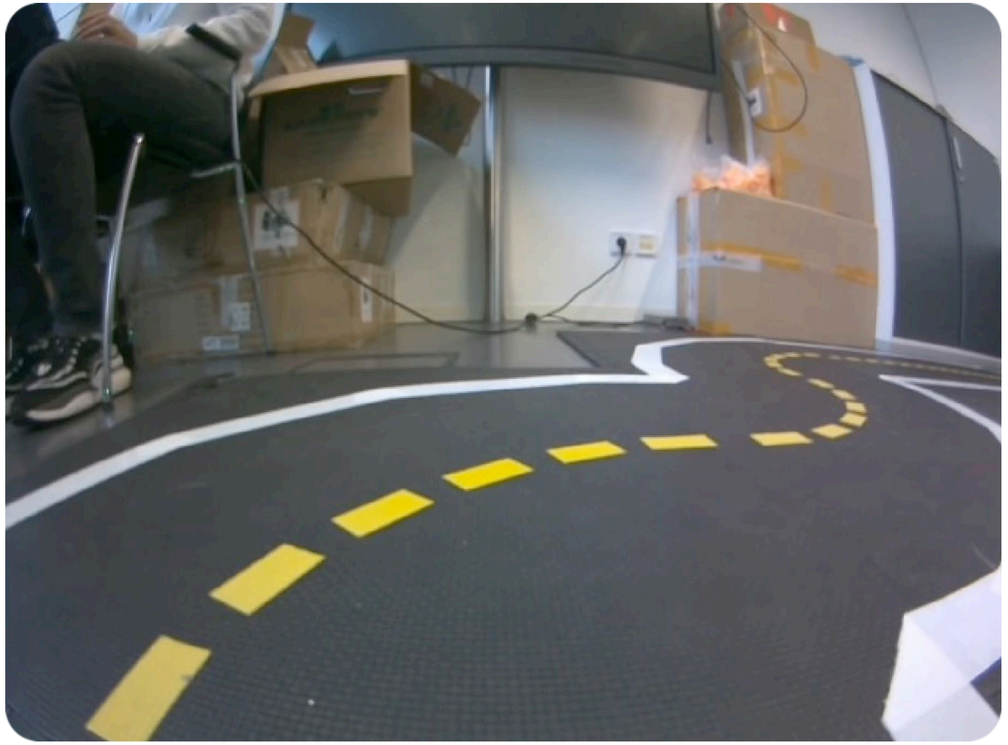
Past

Where did we leave off?

Past

Moving around the city

Using the keypad



Using lane following

Implemented Features

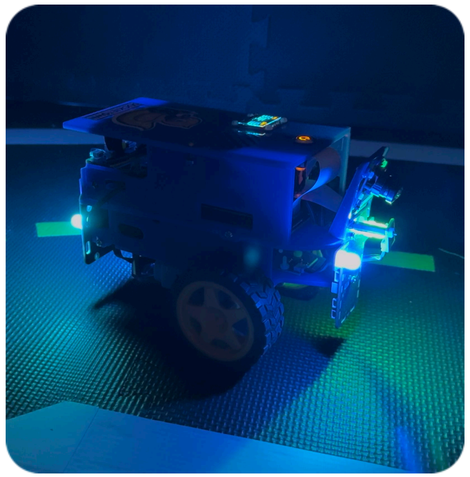
What have we achieved?

Chapter 4.1

FSM States

4.1 FSM States

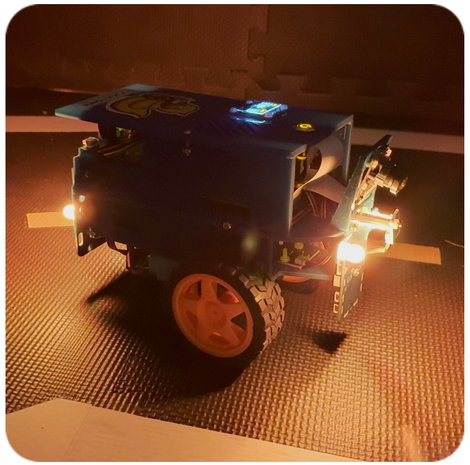
Modes that define robot behaviour



IDLE_MODE

NORMAL_JOYSTICK_CONTROL

LANE_FOLLOWING



RECOVERY_MODE

Chapter 4.2

Lane Following

4.2 Lane Following

Code implementation

lane_controller_custom_node.py

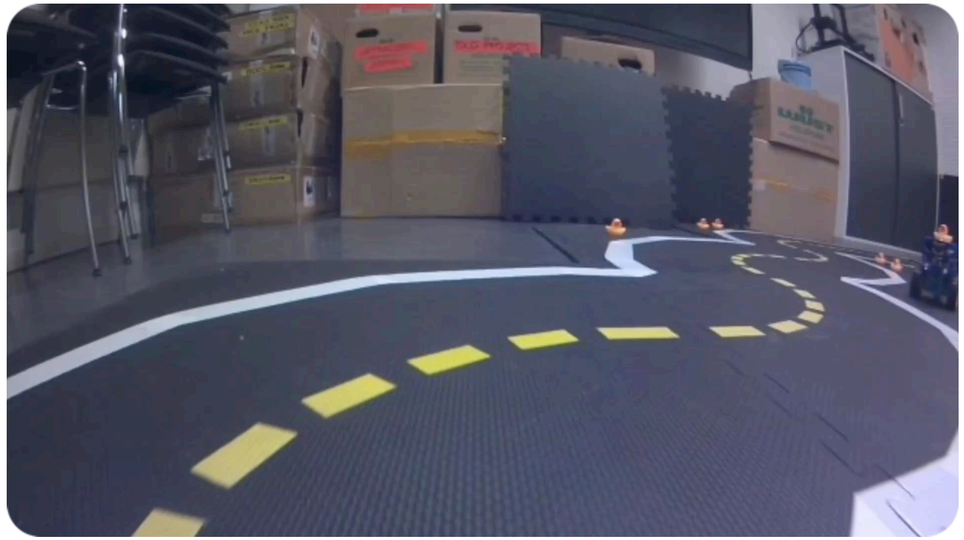
```python

        else:  # Lane following
            # Compute errors
            d_err = pose_msg.d - self.params["~d_offset"].value
            phi_err = pose_msg.phi

            # We cap the error if it grows too large
            if np.abs(d_err) > self.params["~d_thres"].value:
                self.log("d_err too large, thresholding it!", "warn")
                d_err = np.sign(d_err) * self.params["~d_thres"].value

            if phi_err > self.params["~theta_thres_max"].value or phi_err < self.params["~theta_thres_min"].value:
                self.log(f"phi_err too small/large({phi_err}), thresholding it!", "warn")
                phi_err = np.clip(phi_err, self.params["~theta_thres_min"].value, self.params["~theta_thres_max"].value)

            wheels_cmd_exec = [self.wheels_cmd_executed.vel_left, self.wheels_cmd_executed.vel_right]

            if self.obstacle_stop_line_detected:
                v, omega = self.controller.compute_control_action(d_err, phi_err, dt, wheels_cmd_exec, self.obstacle_stop_line_distance)

            else:
                self.stop_line_distance = None
                v, omega = self.controller.compute_control_action(d_err, phi_err, dt, wheels_cmd_exec, self.stop_line_distance)

            # For feedforward action (i.e. during intersection navigation)
            omega += self.params["~omega_ff"].value

            # Initialize car control msg, add header from input message
            car_control_msg.header = pose_msg.header

            # Add commands to car message
            car_control_msg.v = v
            car_control_msg.omega = omega
            self.publish_cmd(car_control_msg)

```

LightEdit mode. Access full IDE

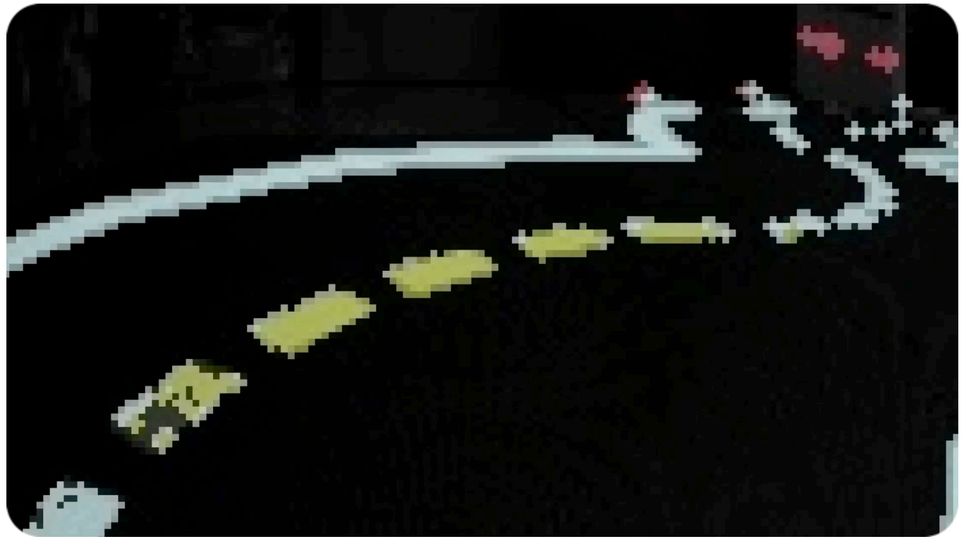
Autosave: off 1:1

4.2 Lane Following

Following the road using computer vision



CAMERA



LINE DETECTOR

Chapter 4.3

Intersection Navigation

4.3 Intersection Navigation

Handling stop lines

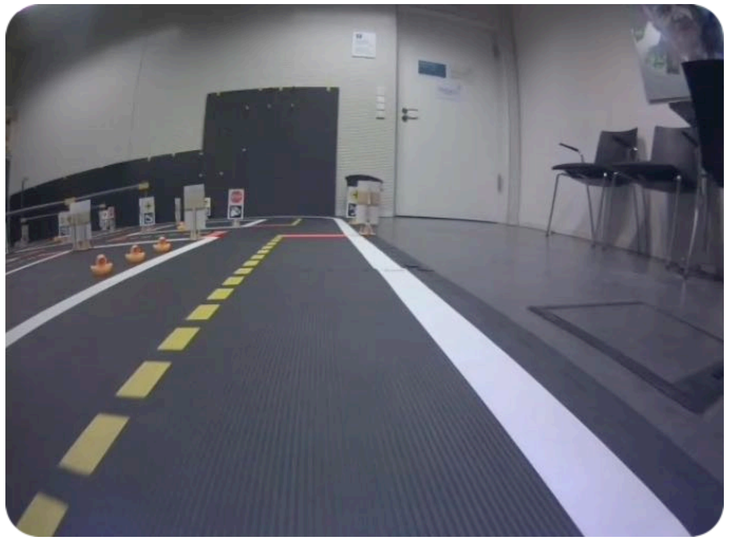
lane_controller_custom_node.py

```python

    def cb_stop_line_reading(self, msg):
        """Callback storing current distance to the next stopline, if one is detected.

        Args:
            msg (:obj:`StopLineReading`): Message containing information about the next stop line.
        """
        # Only stop at stop lines at minimum s second intervals
        if msg.stop_line_detected:
            self.stop_line_distance = np.sqrt(msg.stop_line_point.x**2 + msg.stop_line_point.y**2)
        else:
            self.stop_line_distance = None

        self.at_stop_line = msg.at_stop_line

        if msg.at_stop_line and self.prev_at_stop_line_time is not None:
            if msg.header.stamp.to_sec() - self.prev_at_stop_line_time.to_sec() < 8:
                    return

            self.prev_at_stop_line_time = msg.header.stamp

```

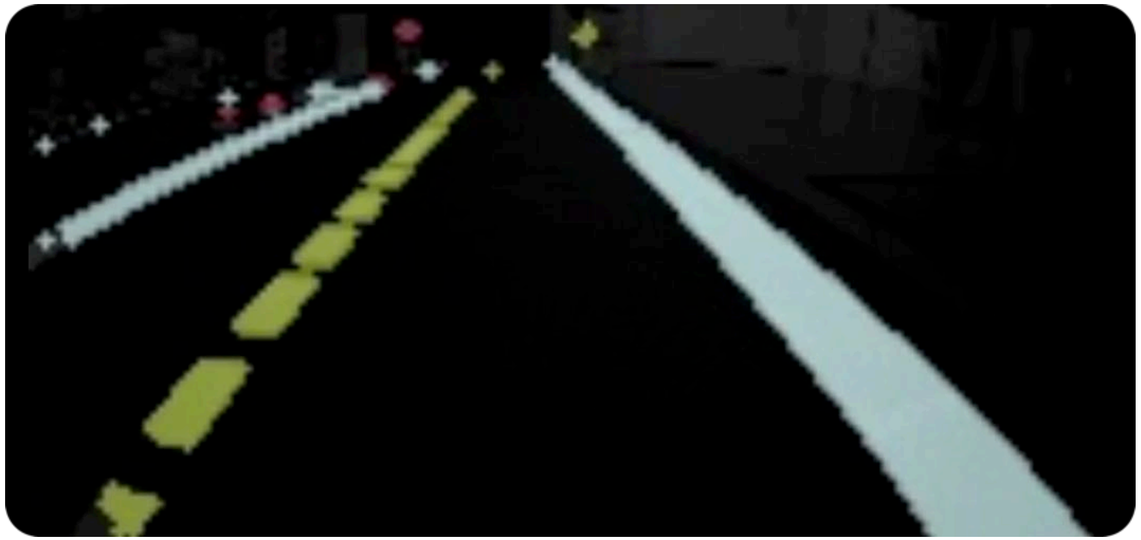
LightEdit mode. Access full IDE Autosave: off 151:41 (16 chars)

4.3 Intersection Navigation

Detecting and stopping at the stop line



CAMERA



LINE DETECTOR

4.3 Intersection Navigation

Handling turns

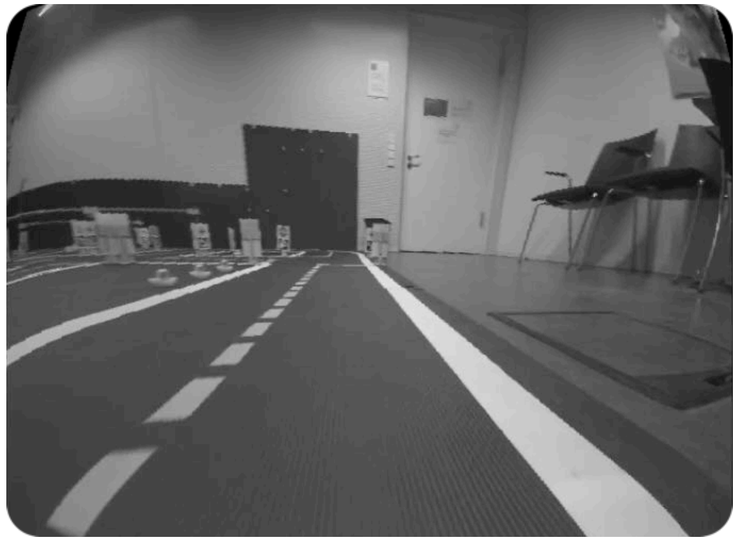
lane_controller_custom_node.py

```python

    def execute_turn(self, turn_type):
        if turn_type == 1:
            return

        self.change_leds(self.led_signals[turn_type])

        # Wait at the stop line
        self.log(f"Sleeping for {self.stop_time.value} seconds")
        rospy.sleep(self.stop_time.value)

        # Construct turning command
        v, omega, sleep_time = self.turn_params[turn_type]
        car_control_msg = Twist2DStamped()
        car_control_msg.header.stamp = rospy.Time.now()
        car_control_msg.v = v.value
        car_control_msg.omega = omega.value

        # Turn
        self.publish_cmd(car_control_msg)
        self.log("Turning now")
        rospy.sleep(sleep_time.value)

        # Construct the turn stopping command
        car_stop_msg = Twist2DStamped()
        car_stop_msg.v = 0
        car_stop_msg.omega = 0

        # Stop turn
        self.publish_cmd(car_stop_msg)
        self.log("Stopping turn")

        self.change_leds(String("CAR_DRIVING"))

```

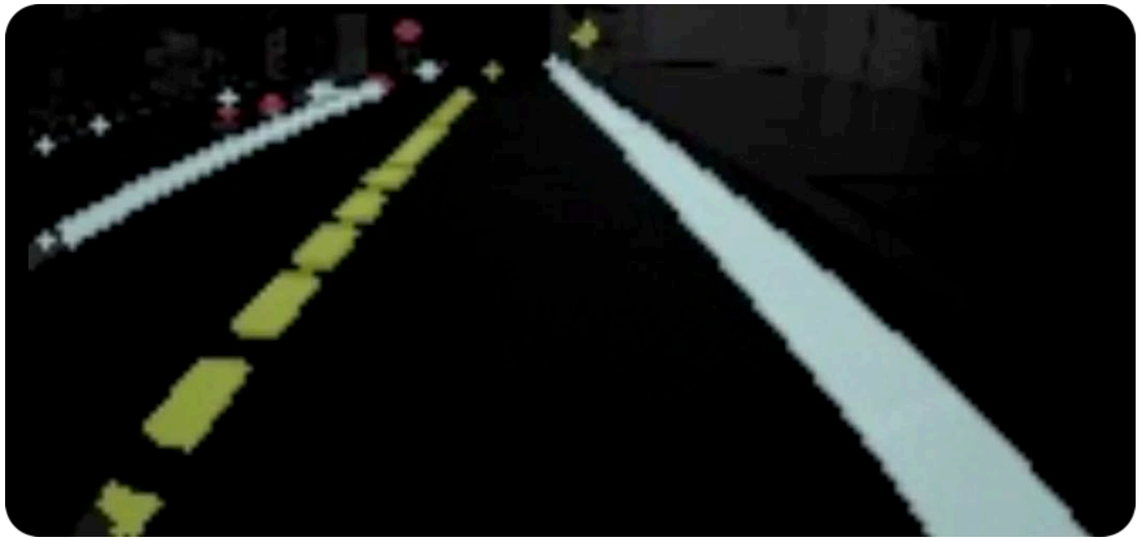
LightEdit mode. Access full IDE Autosave: off 231:57

4.3 Intersection Navigation

Detecting intersections and executing turns



APRILTAG DETECTOR



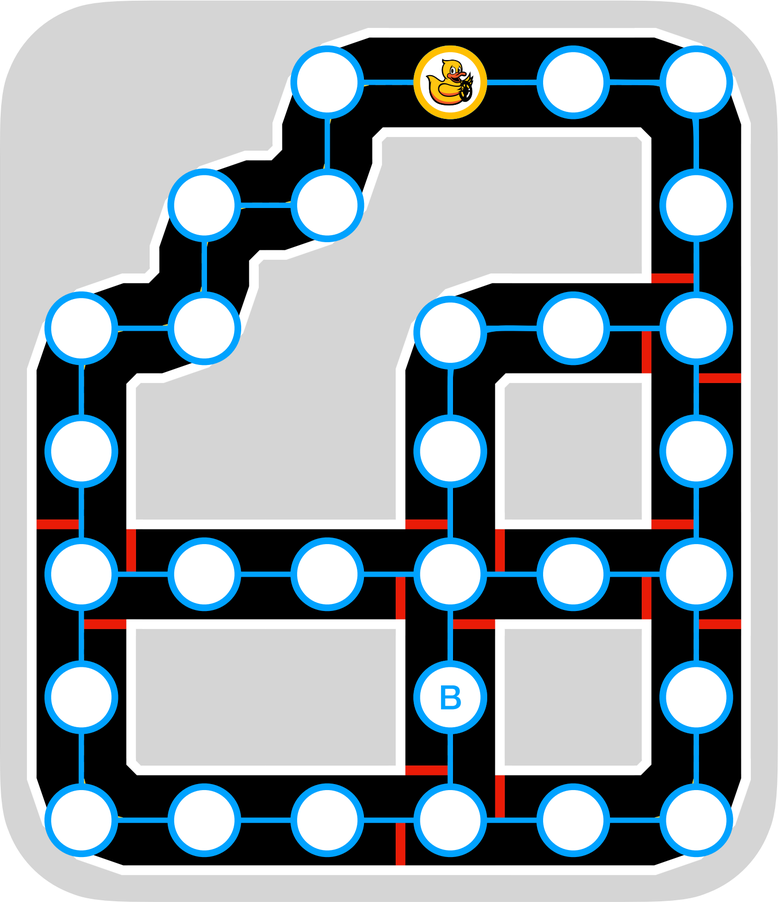
LINE DETECTOR

Chapter 4.4

Navigation using Dijkstra's Algorithm

4.4 Navigation using Dijkstra's Algorithm

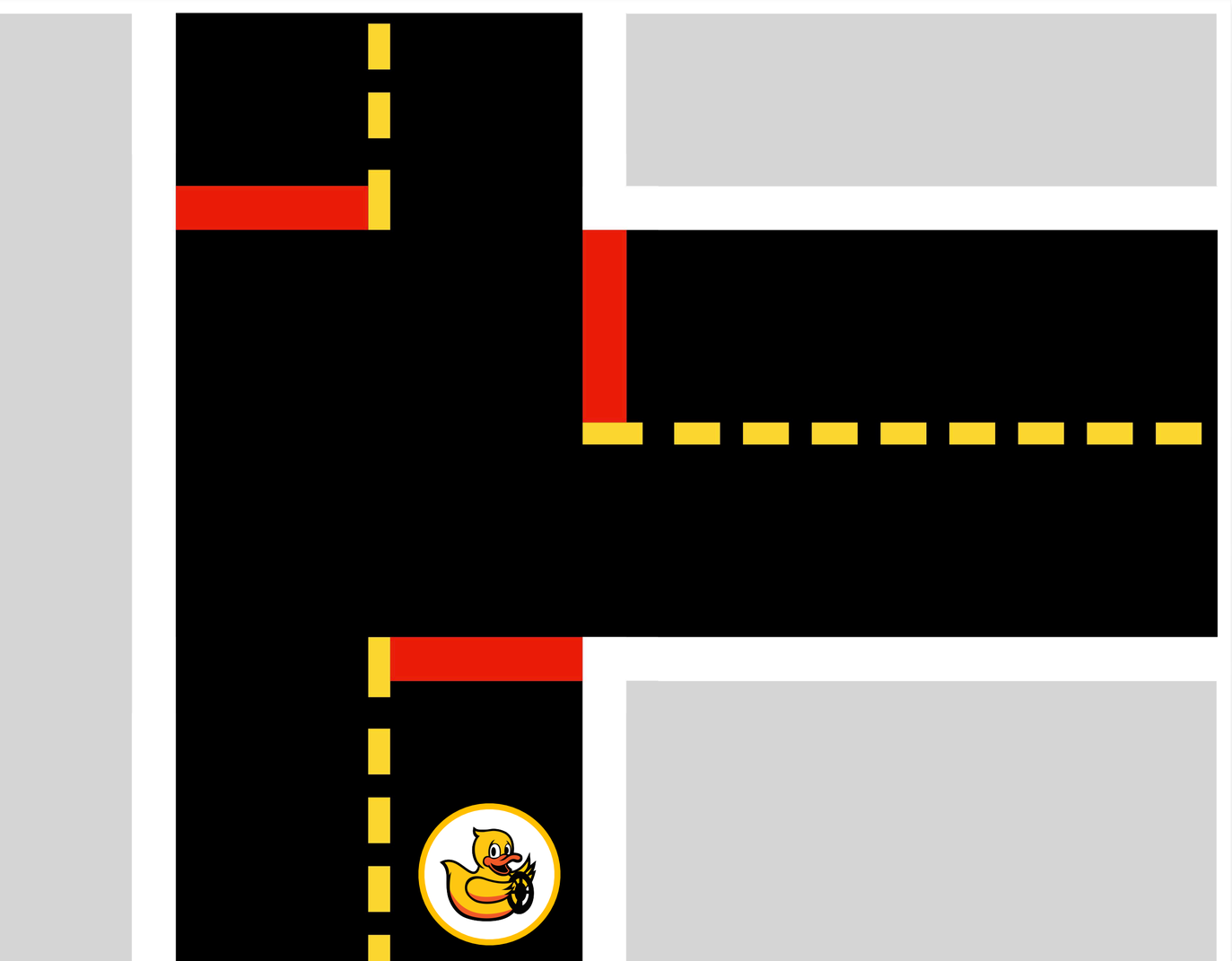
Transforming Roads into a Navigable Network



1. Obtain the Map
2. Define Nodes and Edges
3. Set Points A and B
4. Compute the Shortest Path
5. Navigate

4.4 Navigation using Dijkstra’s Algorithm

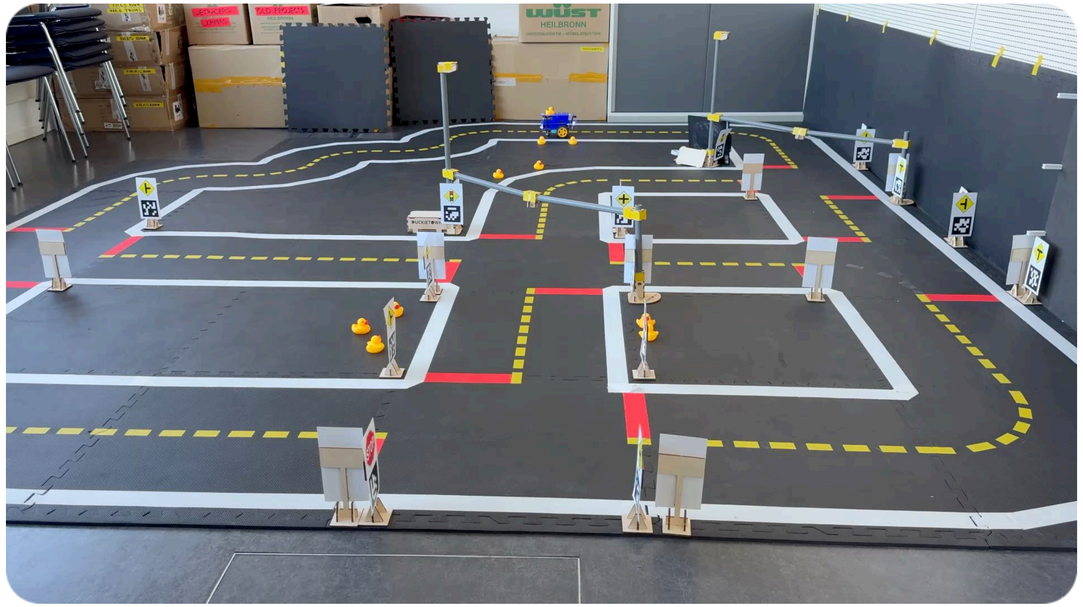
Getting Turn Directions

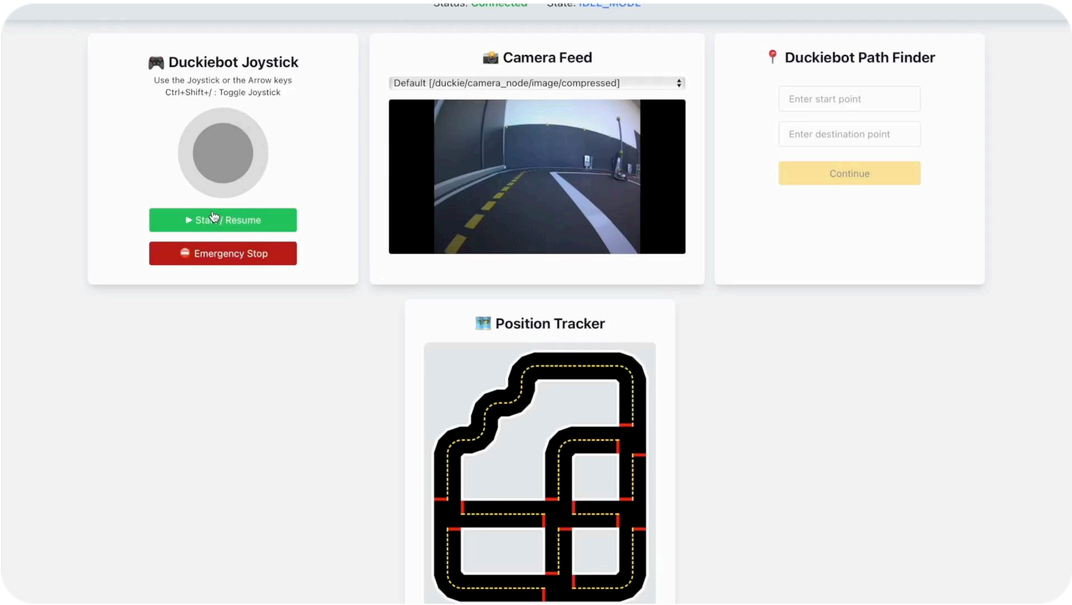


Topics	
lane_controller_node/ intersection_done	**dijkstra_turns_node/ turn_type**
False	Next Turn: Right

4.4 Navigation using Dijkstra’s Algorithm

Demo



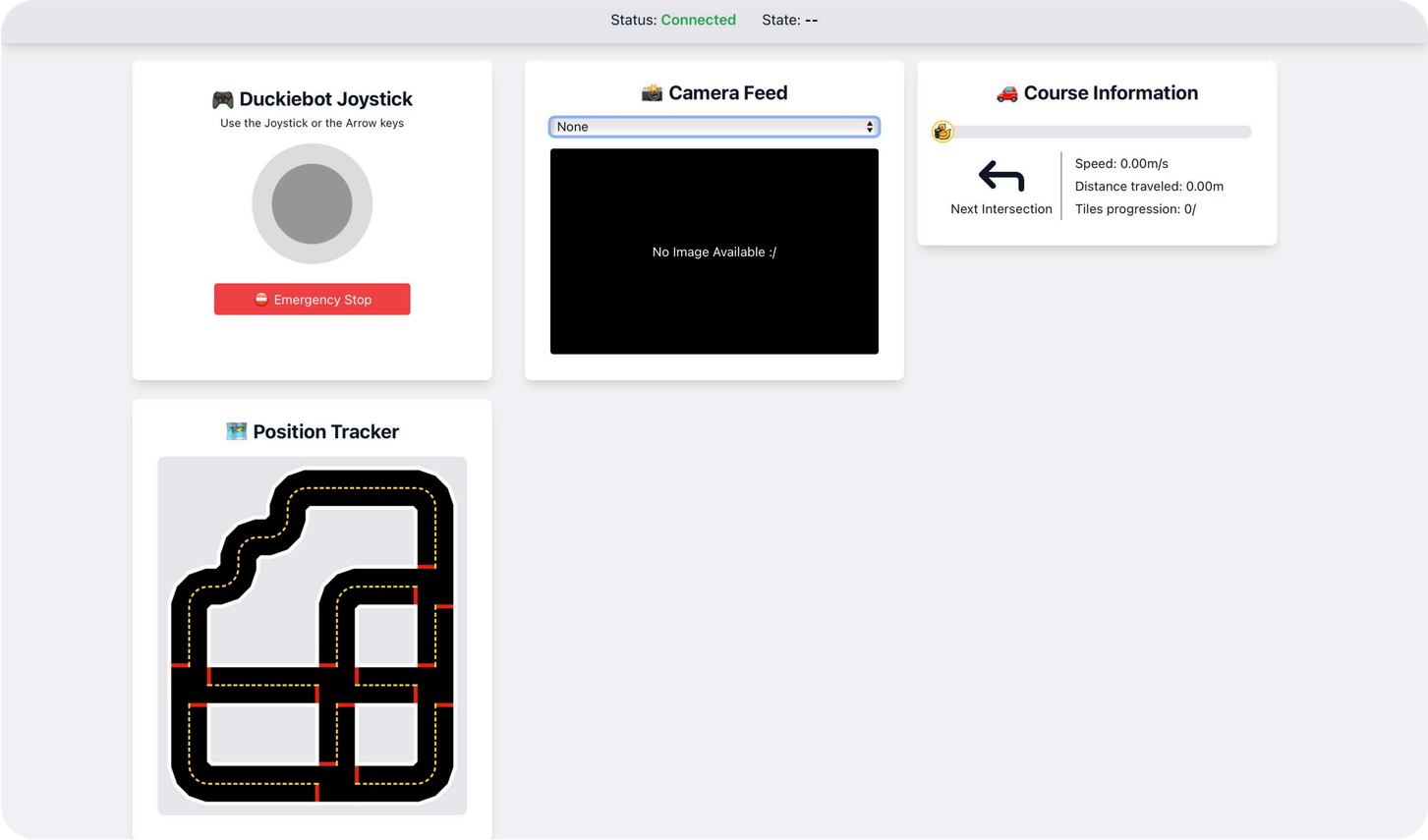


Chapter 4.5

Custom Web Dashboard

4.5 Custom Web Dashboard

Monitor, Control, Navigate



Autonomous Driving Safety

Safety First in Autonomy

Chapter 5.1

Lane-Keeping

5.1 Lane-Keeping

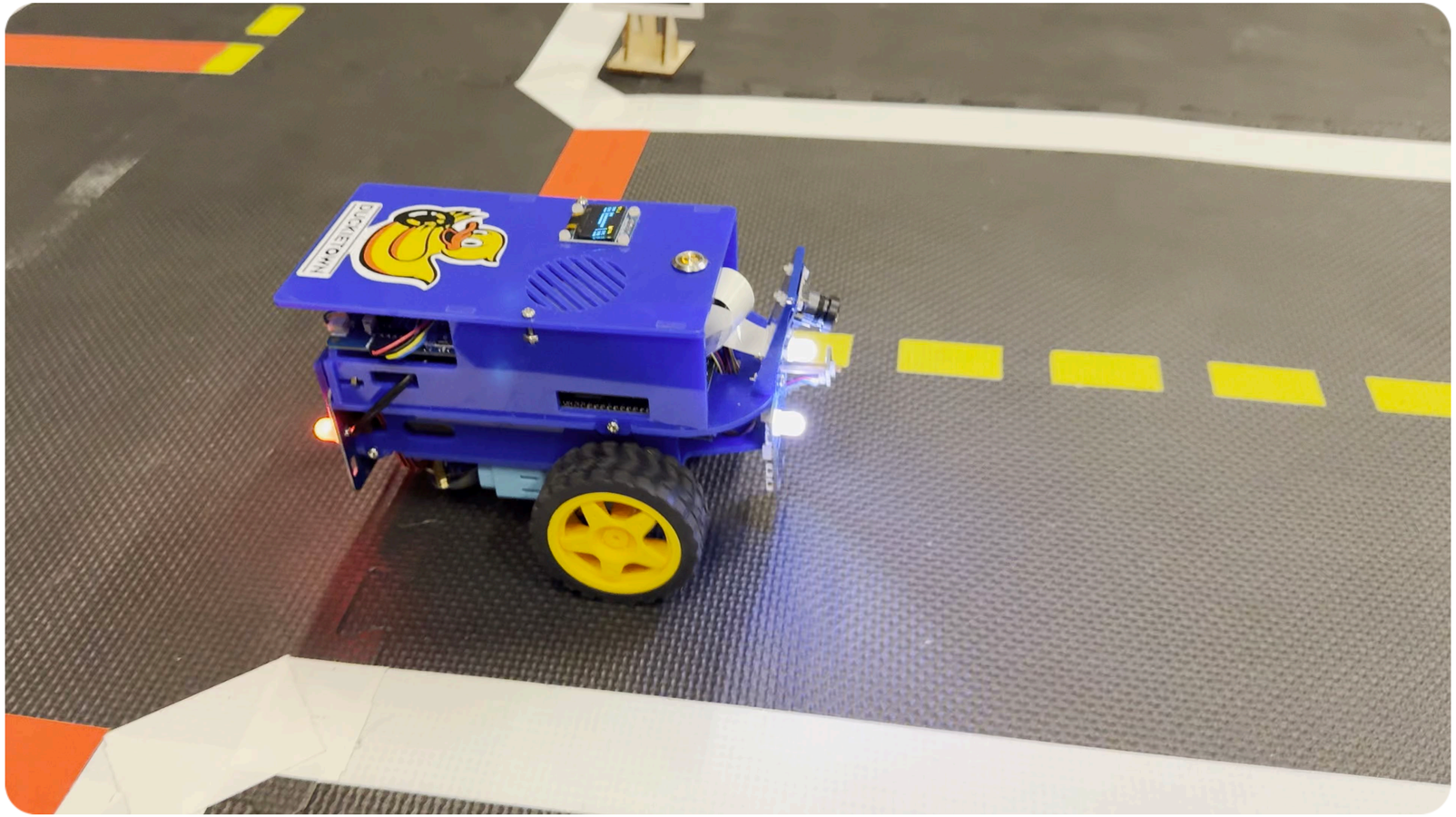
Staying on Track

Chapter 5.2

Vehicle Detection

5.2 Vehicle Detection

Detect, Decide, Avoid



Chapter 5.3

Intersection Safety & Priority Handling

5.3 Intersection Safety & Priority Handling

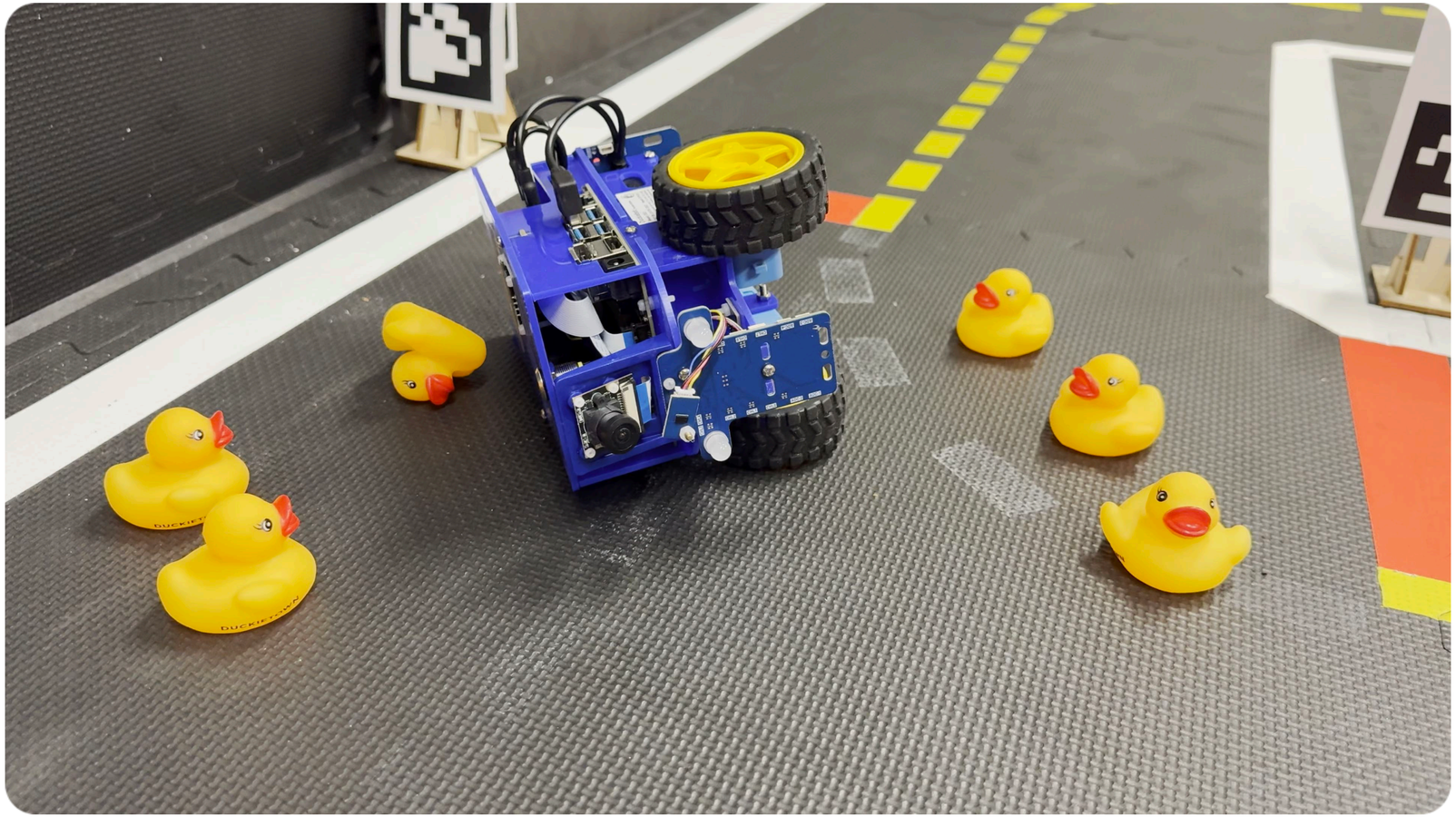
Secure Intersection Handling

Chapter 5.4

Emergency Stop & Fail-Safe Mechanisms

5.4 Emergency Stop & Fail-Safe Mechanisms

Stopping Danger Before It Happens



Conclusion

Final Insights

QuackSquad

Thank you!

Tuesday, 18 March 2025