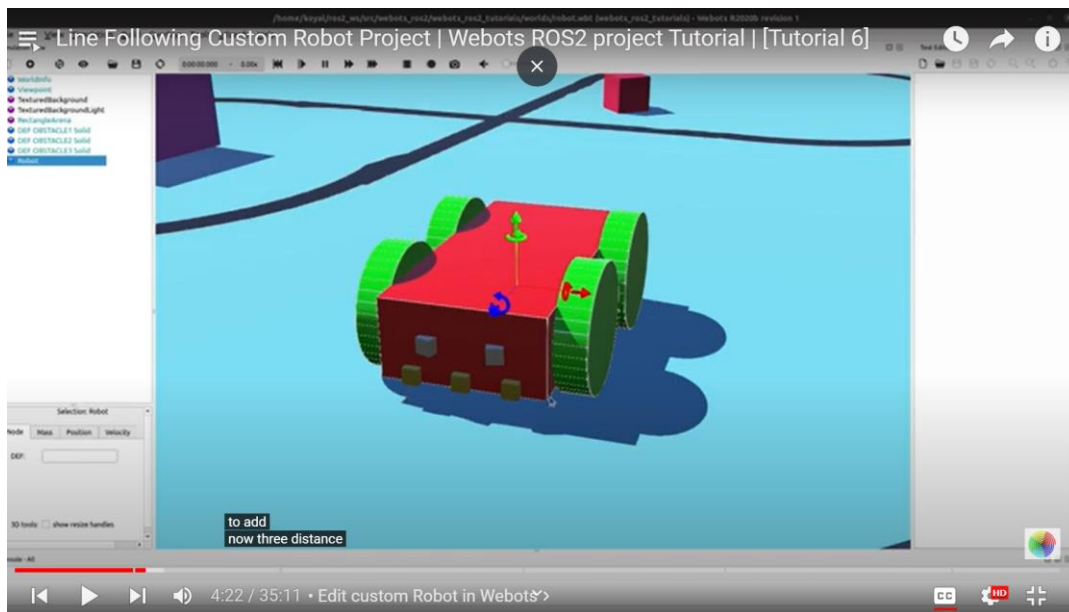


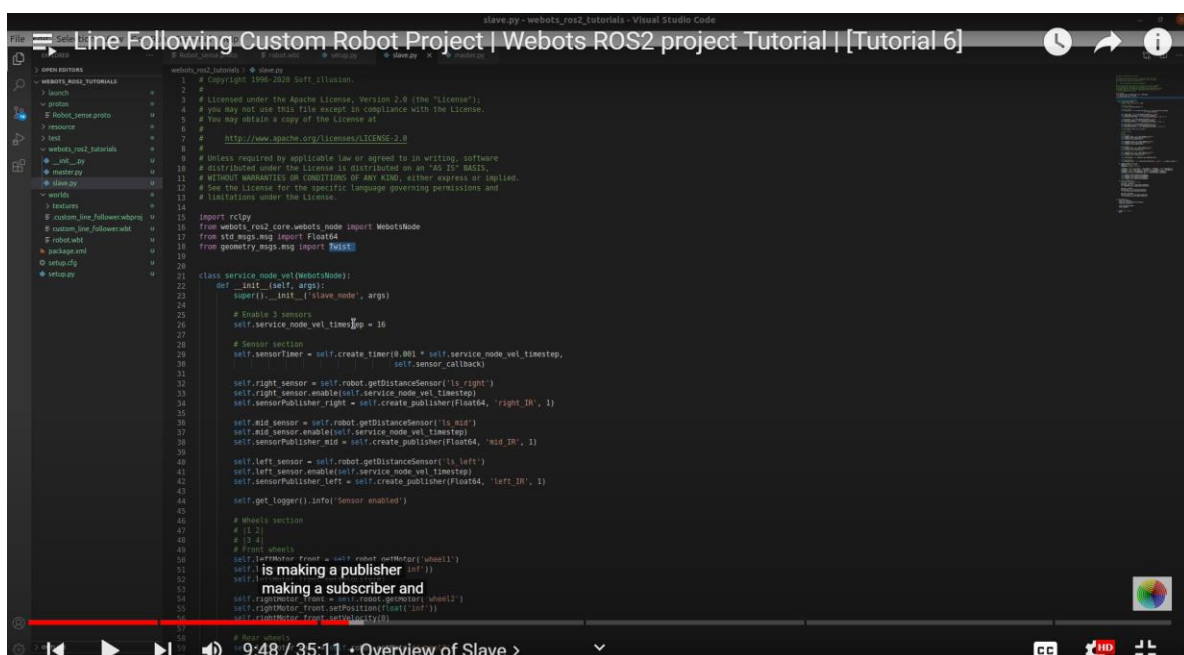
Nama: Dilara Kynta Putri Rafilta
NIM: 1103204059
Kelas: TK44G4

Lecture 5

Pada video tutorial 6 kita mempelajari mengenai Line Following Custom Robot Project. Pada tahap awal akan dijelaskan bagaimana langkah dan flowchart sistem kerja dari robot. Pada project ini terdapat tiga sensor yang akan mendeteksi warna yang membuat robot mengikuti warna garis tersebut.



Langkah pertama adalah membuat file slave.py agar dapat berinteraksi dengan robot. Pada slave.py dilakukan pengaturan terhadap badan roda dan sensor yang ada pada robot serta dapat melakukan pengaturan terhadap jarak sensor terhadap obstacle sesuai keinginan.



Pada node master.py , sensor akan mengatur direction robot pergi dan akan dipublish ke comman velocities.

```

1 # Copyright 1996-2020 Soft Illusion.
2 #
3 # Licensed under the Apache License, Version 2.0 (the "License");
4 # you may not use this file except in compliance with the License.
5 # You may obtain a copy of the License at
6 # http://www.apache.org/licenses/LICENSE-2.0
7 #
8 # Unless required by applicable law or agreed to in writing, software
9 # distributed under the License is distributed on an "AS IS" BASIS,
10 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
11 # See the License for the specific language governing permissions and
12 # limitations under the License.
13
14 import rclpy
15 from rclpy.node import Node
16 from std_msgs.msg import Float64
17 from geometry_msgs.msg import Twist
18
19 class LineFollower(Node):
20     def __init__(self):
21         super().__init__('linefollower_cmdvel')
22         # Subscribe to sensor
23         self.subs_right_ir = self.create_subscription(Float64, 'right_IR', self.rightIR_cb, 1)
24         self.subs_left_ir = self.create_subscription(Float64, 'left_IR', self.leftIR_cb, 1)
25         self.subs_mid_ir = self.create_subscription(Float64, 'mid_IR', self.midIR_cb, 1)
26         # Publish cmd vel
27         self.pubs_cmdvel = self.create_publisher(Twist, 'cmd_vel', 1)
28
29         # Vehicle parameters
30         self.speed = 0.2
31         self.angle_correction = 0.01
32
33         # Initialize parameters
34         self.GS_RIGHT = self.get_parameter('GS_RIGHT').get_float_value()
35         self.GS_LEFT = self.get_parameter('GS_LEFT').get_float_value()
36         self.delay = 0
37         self.cmd = Twist()
38         self.stop = False
39         self.count = 0
40         self.count_threshold = 10
41
42     def lineFollowingModule(self):
43         # Call back to update sensor reading variables
44         self.rightIR_cb(self.rightIR)
45         self.leftIR_cb(self.leftIR)
46         self.midIR_cb(self.midIR)
47
48         # will do
49         # so pretty much this node
50
51     def main(args=None):
52         rclpy.init(args=args)
53         lf = LineFollower()
54         lf.create_timer(0.1, lf.lineFollowingModule)
55         lf.destroy_node()
56         rclpy.shutdown()
57
58 if __name__ == '__main__':
59     main()

```

Selanjutnya pada node line_following_launch.py akan dilakukan penyimpanan packages directory yang telah dibuat di langkah-langkah sebelumnya serta melakukan integrasi antar file line following dengan setup.py

```

1 # Copyright 1996-2020 Soft Illusion.
2 #
3 # Licensed under the Apache License, Version 2.0 (the "License");
4 # you may not use this file except in compliance with the License.
5 # You may obtain a copy of the License at
6 # http://www.apache.org/licenses/LICENSE-2.0
7 #
8 # Unless required by applicable law or agreed to in writing, software
9 # distributed under the License is distributed on an "AS IS" BASIS,
10 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
11 # See the License for the specific language governing permissions and
12 # limitations under the License.
13
14 import rclpy
15 from rclpy.node import Node
16 from std_msgs.msg import Float64
17 from geometry_msgs.msg import Twist
18
19 class LineFollower(Node):
20     def __init__(self):
21         super().__init__('linefollower_cmdvel')
22         # Subscribe to sensor
23         self.subs_right_ir = self.create_subscription(Float64, 'right_IR', self.rightIR_cb, 1)
24         self.subs_left_ir = self.create_subscription(Float64, 'left_IR', self.leftIR_cb, 1)
25         self.subs_mid_ir = self.create_subscription(Float64, 'mid_IR', self.midIR_cb, 1)
26         # Publish cmd vel
27         self.pubs_cmdvel = self.create_publisher(Twist, 'cmd_vel', 1)
28
29         # Vehicle parameters
30         self.speed = 0.2
31         self.angle_correction = 0.01
32
33         # Initialize parameters
34         self.GS_RIGHT = self.get_parameter('GS_RIGHT').get_float_value()
35         self.GS_LEFT = self.get_parameter('GS_LEFT').get_float_value()
36         self.delay = 0
37         self.cmd = Twist()
38         self.stop = False
39         self.count = 0
40         self.count_threshold = 10
41
42     def lineFollowingModule(self):
43         # Call back to update sensor reading variables
44         self.rightIR_cb(self.rightIR)
45         self.leftIR_cb(self.leftIR)
46         self.midIR_cb(self.midIR)
47
48         # will do
49         # so pretty much this node
50
51     def main(args=None):
52         rclpy.init(args=args)
53         lf = LineFollower()
54         lf.create_timer(0.1, lf.lineFollowingModule)
55         lf.destroy_node()
56         rclpy.shutdown()
57
58 if __name__ == '__main__':
59     main()

```

Langkah terakhir adalah melakukan build and demo untuk project yang telah di set up

