

Nama : Nabil Putra Kurnia Utomo

Lecture Week 5

Report Week5 Video 6

Untuk Video 6, tidak tersedia tutorial file sebagaimana yang diilustrasikan dalam video tersebut. Namun, langkah-langkah pertama melibatkan pengeditan manual pada model robot menggunakan Webots. Selanjutnya, perlu dibuat file `slave.py` menggunakan Visual Studio Code untuk mengonfigurasi sensor kanan, kiri, dan parameter lain pada robot. Pastikan pembuatan file ini dilakukan dengan hati-hati untuk menghindari kesalahan. Arah dan kecepatan rotasi roda robot juga diatur di bagian ini, bersamaan dengan pembatasan kecepatannya.

File Edit Selection View Go Run Terminal Help

robot_setup.py

```

1 # Copyright 1996-2008 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 #
7 # http://www.apache.org/licenses/LICENSE-2.0
8 #
9 # Unless required by applicable law or agreed to in writing, software
10 # distributed under the License is distributed on an "AS IS" BASIS,
11 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 # See the License for the specific language governing permissions and
13 # limitations under the License.
14
15 import rcipy
16 from webots_ros2_core.webots_node import WebotsNode
17 from std_msgs.msg import Float64
18 from geometry_msgs.msg import Twist
19
20
21 class service_node_vel(WebotsNode):
22     def __init__(self, args):
23         super().__init__('service_node', args)
24
25         # Enable 3 sensors
26         self.service_node_vel_timestep = 16
27
28         # Sensor section
29         self.sensorTimer = self.create_timer(0.001 * self.service_node_vel_timestep,
30                                             self.sensor_callback)
31
32         self.right_sensor = self.robot.getDistanceSensor('ls_right')
33         self.right_sensor.enable(self.service_node_vel_timestep)
34         self.sensorPublisher_right = self.create_publisher(Float64, 'right_ID', 1)
35
36         self.mid_sensor = self.robot.getDistanceSensor('ls_mid')
37         self.mid_sensor.enable(self.service_node_vel_timestep)
38         self.sensorPublisher_mid = self.create_publisher(Float64, 'mid_ID', 1)
39
40         self.left_sensor = self.robot.getDistanceSensor('ls_left')
41         self.left_sensor.enable(self.service_node_vel_timestep)
42         self.sensorPublisher_left = self.create_publisher(Float64, 'left_ID', 1)
43
44         self.get_logger().info('Sensor enabled')
45
46         # Wheels section
47         # [ 1 2 ]
48         # [ 3 4 ]
49         # Front wheels
50         self.leftMotor_front = self.robot.getMotor('wheel1')
51         self.leftMotor_front.setPosition(float('inf'))
52         self.leftMotor_front.setVelocity(0)
53
54         self.rightMotor_front = self.robot.getMotor('wheel2')
55         self.rightMotor_front.setPosition(float('inf'))
56         self.rightMotor_front.setVelocity(0)
57
58         # Rear wheels
59         self.leftMotor_rear = self.robot.getMotor('wheel3')
60         self.leftMotor_rear.setPosition(float('inf'))

```

Langkah berikutnya adalah membuat file master.py, yang berlangganan (subscribe) terhadap pesan-pesan sensor yang dipublikasikan oleh node slave. File ini juga membuat perkiraan arah dan mempublikasikan pesan Twist ke topik /cmd_vel menggunakan ROS2. Properti pencahayaan digunakan di sini, dan berbagai warna memiliki pengaruh berbeda. Kecepatan linear diatur sebesar 0,2. Jika terdapat perbedaan nilai antara sensor kiri dan kanan, itu menandakan robot bergerak dalam arah yang tidak benar karena salah satu sensor melihat warna hitam. Oleh karena itu, perbedaan ini dikalikan dengan variabel koreksi sudut untuk mengarahkan robot. Jika sensor tengah tidak mendeteksi garis hitam, logika ditulis untuk menghentikan robot setelah beberapa saat. Di sini, kecepatan linear dan angular diberikan nilai 0.

Setelahnya, robot di-setup dan dapat diluncurkan menggunakan perintah `colcon build proyek`. Proyek kemudian dapat dijalankan dengan perintah `ros2 launch webots_ros2_tutorial line_following.launch`. Perintah `ros2 topic list` akan menampilkan daftar semua topik yang aktif. Setelah semuanya selesai, robot seharusnya dapat mengikuti garis hitam dengan benar dan melakukan koreksi jika diperlukan. Jika robot dipindahkan dari garis hitam, robot akan berhenti bergerak.

Pastikan untuk mengikuti setiap langkah dengan cermat dan memastikan bahwa setiap perintah dan konfigurasi dilakukan dengan benar untuk mencapai hasil yang diinginkan.