

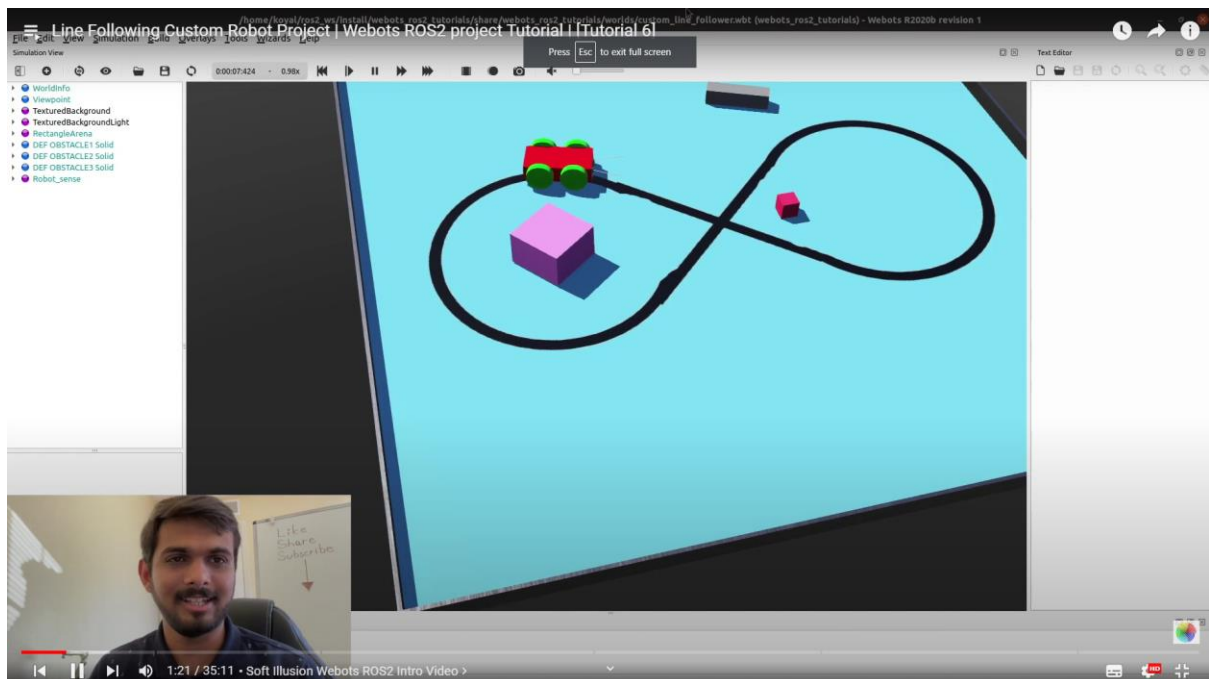
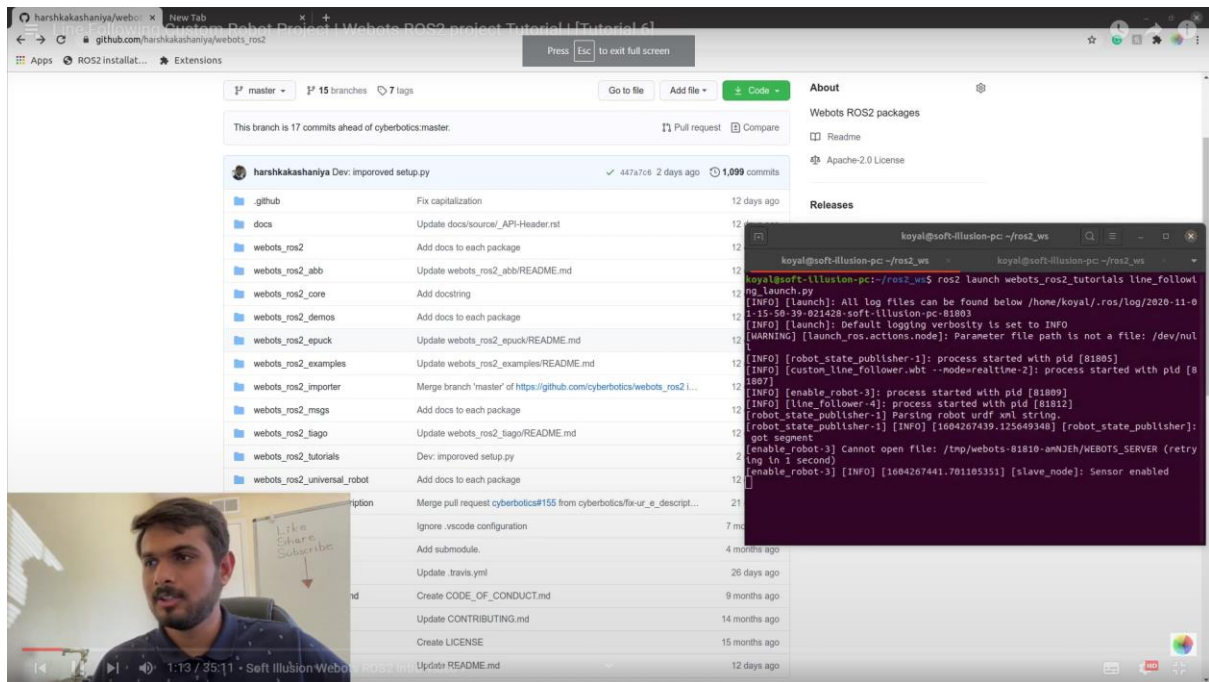
Nama : Ichlasul Al Gifari

NIM : 1103204236

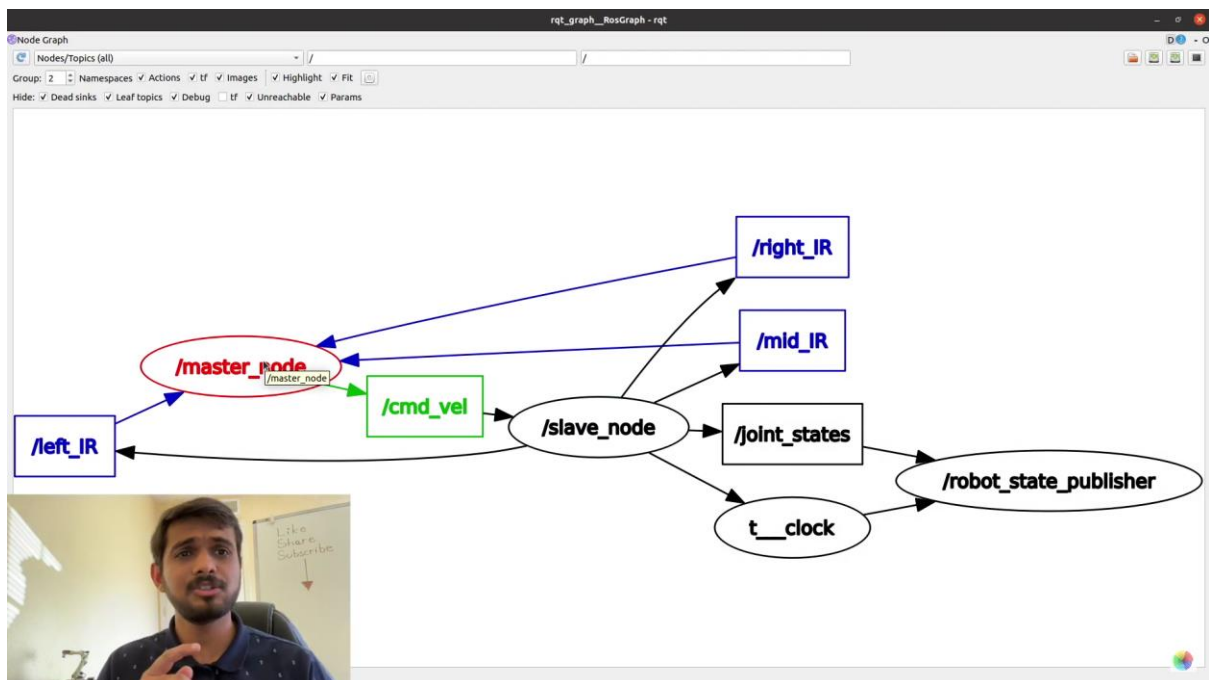
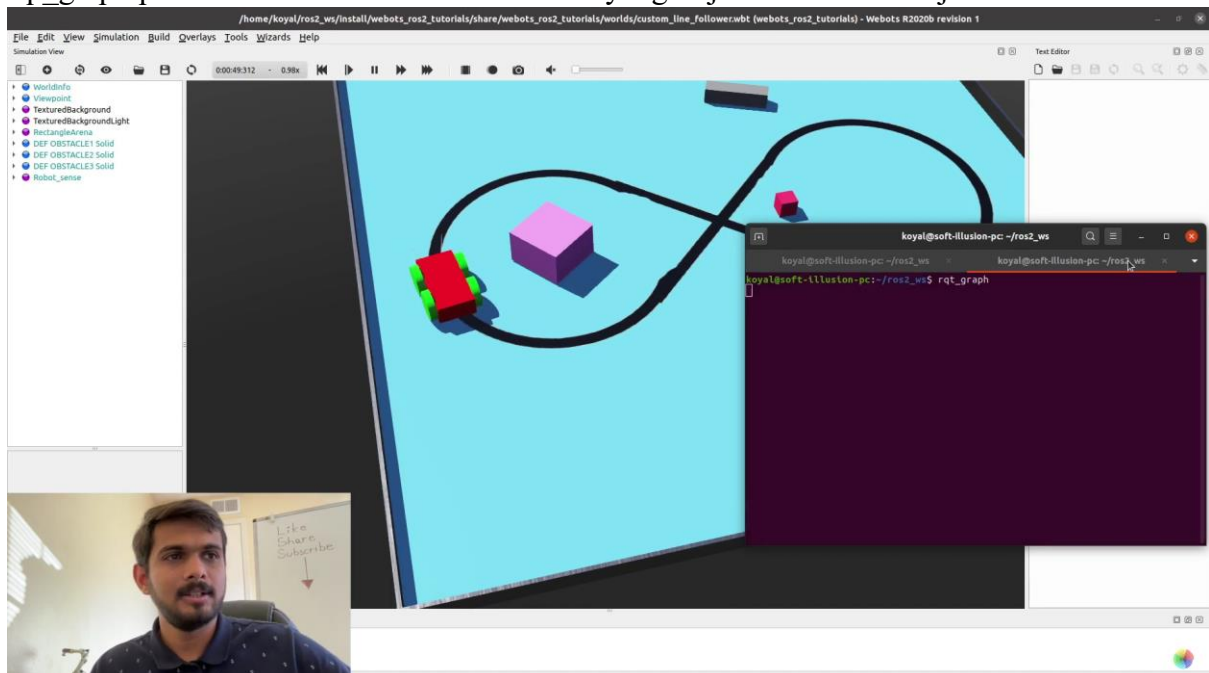
## Tugas Robotik Lecture 5

### Tahap 1

Setelah menginstall dan menjalankan webots selanjutnya memasukkan perintah untuk menjalankan simulasi robot yang sudah diimplementasikan pada video sebelumnya.

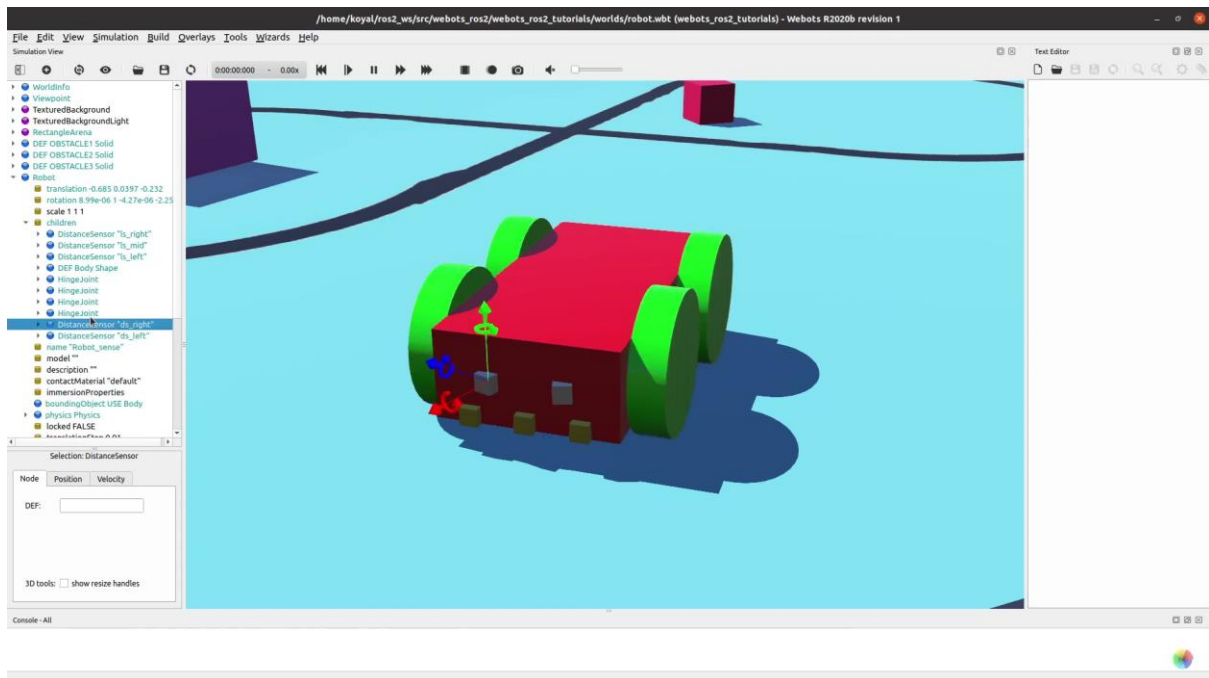


Setelah melihat simulasi robot berjalan dengan baik, lalu masukkan perintah `rqt_graph` pada terminal untuk melihat urutan yang terjadi saat robot berjalan.

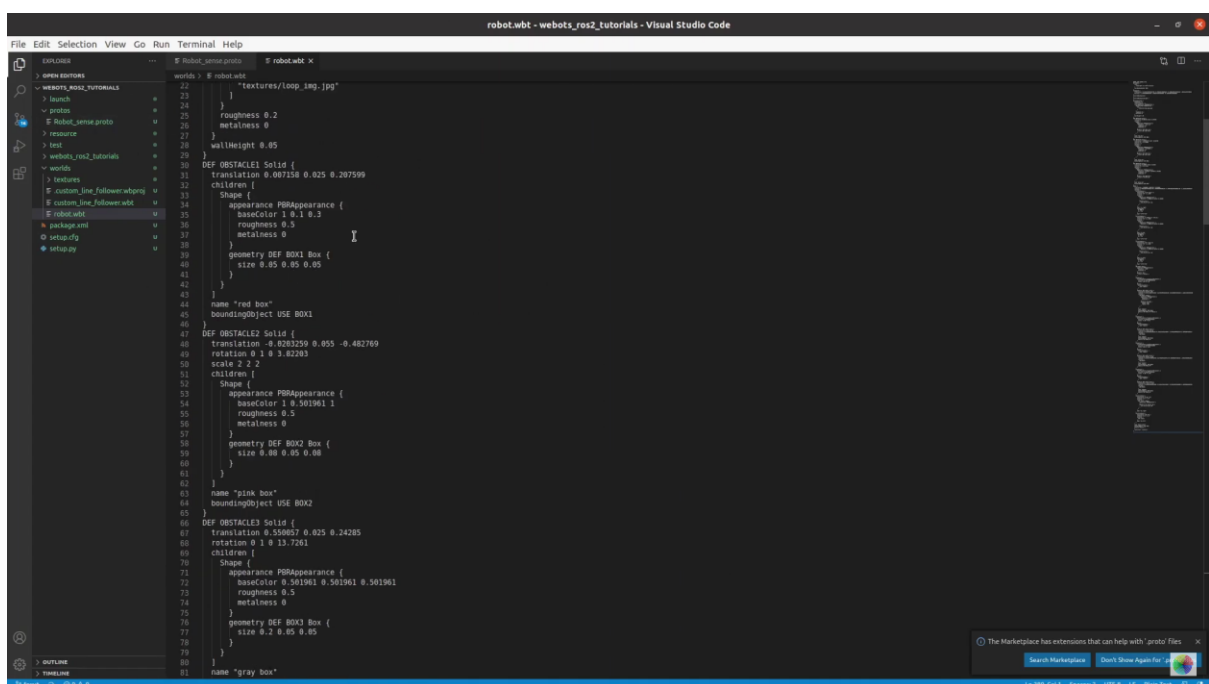


## Tahap 2

Selanjutnya, setelah menjalankan simulasi kita bisa melakukan modifikasi pada robot yang kita punya.



Setelah memodifikasi bentuk robot seperti bagian body, roda, dan bentuk lalu disimpan hasil modifikasinya kita bisa melihat hasil modifikasi tersebut pada visual code.



### Tahap 3

Setelah memasuki visual code, kita dapat meriview semua yang sudah kita lakukan dengan robot dan webots kita.

slave.py - webots\_ros2\_tutorials - Visual Studio Code

```
1 # Copyright 1996-2020 Soft Robotics.
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 rclpy
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 class ServiceNodeVel(WebotsNode):
21     def __init__(self, args):
22         super().__init__('slave_node', args)
23
24         # Enable 3 sensors
25         self.service_node_vel_timestep = 16
26
27         # Sensor section
28         self.sensor_timer = self.create_timer(0.001 * self.service_node_vel_timestep,
29                                               self.sensor_callback)
30
31         self.right_sensor = self.robot.getDistanceSensor('ls_right')
32         self.right_sensor.enable(self.service_node_vel_timestep)
33         self.sensor_publisher_right = self.create_publisher(Float64, 'right_3R', 1)
34
35         self.mid_sensor = self.robot.getDistanceSensor('ls_mid')
36         self.mid_sensor.enable(self.service_node_vel_timestep)
37         self.sensor_publisher_mid = self.create_publisher(Float64, 'mid_3R', 1)
38
39         self.left_sensor = self.robot.getDistanceSensor('ls_left')
40         self.left_sensor.enable(self.service_node_vel_timestep)
41         self.sensor_publisher_left = self.create_publisher(Float64, 'left_3R', 1)
42
43         self.get_logger().info('Sensor enabled')
44
45         # Wheels section
46         # [ 2 ]
47         # [ 3 4 ]
48
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'))
```

slave.py - webots\_ros2\_tutorials - Visual Studio Code

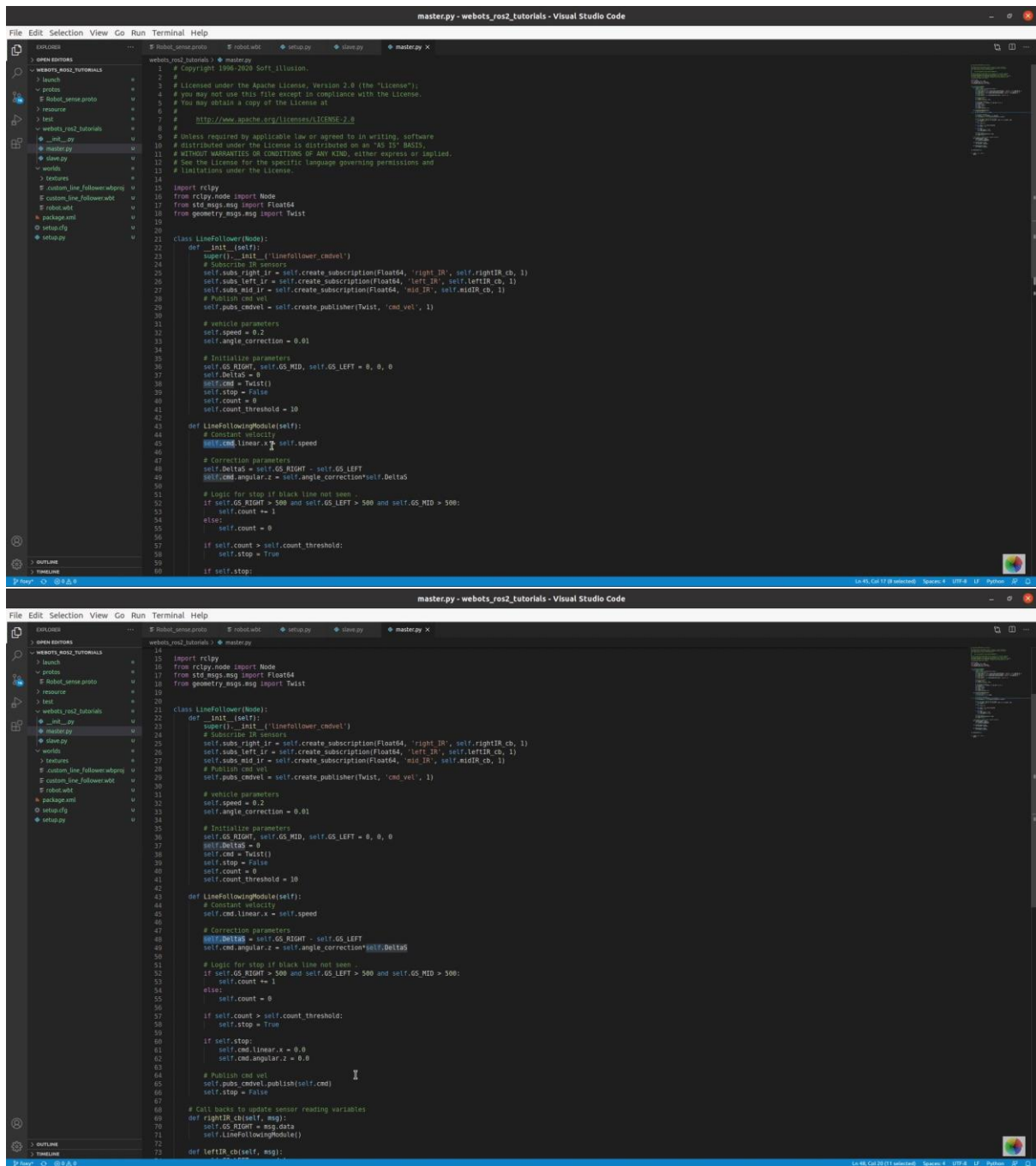
```
1 # Copyright 1996-2020 Soft Robotics.
2 #
3 # Licensed under the Apache License, Version 2.0 (the "License");
4 # you may not use this file except in compliance with the License.
5 # You may obtain a copy of the License at
6 #
7 # http://www.apache.org/licenses/LICENSE-2.0
8 #
9 # Unless required 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 rclpy
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 class ServiceNodeVel(WebotsNode):
21     def __init__(self, args):
22         super().__init__('slave_node', args)
23
24         # Enable 3 sensors
25         self.service_node_vel_timestep = 16
26
27         # Sensor section
28         self.sensor_timer = self.create_timer(0.001 * self.service_node_vel_timestep,
29                                               self.sensor_callback)
30
31         self.right_sensor = self.robot.getDistanceSensor('ls_right')
32         self.right_sensor.enable(self.service_node_vel_timestep)
33         self.sensor_publisher_right = self.create_publisher(Float64, 'right_3R', 1)
34
35         self.mid_sensor = self.robot.getDistanceSensor('ls_mid')
36         self.mid_sensor.enable(self.service_node_vel_timestep)
37         self.sensor_publisher_mid = self.create_publisher(Float64, 'mid_3R', 1)
38
39         self.left_sensor = self.robot.getDistanceSensor('ls_left')
40         self.left_sensor.enable(self.service_node_vel_timestep)
41         self.sensor_publisher_left = self.create_publisher(Float64, 'left_3R', 1)
42
43         self.get_logger().info('Sensor enabled')
44
45         # Wheels section
46         # [ 2 ]
47         # [ 3 4 ]
48
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'))
61
62     def sensor_callback(self):
63         # Read right sensor
64         right_pos = self.right_sensor.getPosition()
65         self.sensor_publisher_right.publish(Float64(right_pos))
66
67         # Read mid sensor
68         mid_pos = self.mid_sensor.getPosition()
69         self.sensor_publisher_mid.publish(Float64(mid_pos))
70
71         # Read left sensor
72         left_pos = self.left_sensor.getPosition()
73         self.sensor_publisher_left.publish(Float64(left_pos))
74
75     def __del__(self):
76         self.get_logger().info('ServiceNodeVel destroyed')
77
78     def __str__(self):
79         return 'ServiceNodeVel'
80
81     def __repr__(self):
82         return 'ServiceNodeVel'
83
84     def __eq__(self, other):
85         return self.__str__() == other.__str__()
86
87     def __neq__(self, other):
88         return self.__str__() != other.__str__()
89
90     def __lt__(self, other):
91         return self.__str__() < other.__str__()
92
93     def __gt__(self, other):
94         return self.__str__() > other.__str__()
95
96     def __le__(self, other):
97         return self.__str__() <= other.__str__()
98
99     def __ge__(self, other):
100         return self.__str__() >= other.__str__()
101
102     def __add__(self, other):
103         return self.__str__() + other.__str__()
104
105     def __sub__(self, other):
106         return self.__str__() - other.__str__()
107
108     def __mul__(self, other):
109         return self.__str__() * other.__str__()
110
111     def __div__(self, other):
112         return self.__str__() / other.__str__()
113
114     def __mod__(self, other):
115         return self.__str__() % other.__str__()
116
117     def __pow__(self, other):
118         return self.__str__() ** other.__str__()
119
120     def __call__(self):
121         return self.__str__()
122
123     def __getitem__(self, item):
124         return self.__str__[item]
125
126     def __setitem__(self, item, value):
127         self.__str__[item] = value
128
129     def __delitem__(self, item):
130         del self.__str__[item]
131
132     def __len__(self):
133         return len(self.__str__)
134
135     def __iter__(self):
136         return iter(self.__str__)
137
138     def __contains__(self, item):
139         return item in self.__str__
140
141     def __getitem__(self, item):
142         return self.__str__[item]
143
144     def __setitem__(self, item, value):
145         self.__str__[item] = value
146
147     def __delitem__(self, item):
148         del self.__str__[item]
149
150     def __len__(self):
151         return len(self.__str__)
152
153     def __iter__(self):
154         return iter(self.__str__)
155
156     def __contains__(self, item):
157         return item in self.__str__
158
159     def __getitem__(self, item):
160         return self.__str__[item]
161
162     def __setitem__(self, item, value):
163         self.__str__[item] = value
164
165     def __delitem__(self, item):
166         del self.__str__[item]
167
168     def __len__(self):
169         return len(self.__str__)
170
171     def __iter__(self):
172         return iter(self.__str__)
173
174     def __contains__(self, item):
175         return item in self.__str__
176
177     def __getitem__(self, item):
178         return self.__str__[item]
179
180     def __setitem__(self, item, value):
181         self.__str__[item] = value
182
183     def __delitem__(self, item):
184         del self.__str__[item]
185
186     def __len__(self):
187         return len(self.__str__)
188
189     def __iter__(self):
190         return iter(self.__str__)
191
192     def __contains__(self, item):
193         return item in self.__str__
194
195     def __getitem__(self, item):
196         return self.__str__[item]
197
198     def __setitem__(self, item, value):
199         self.__str__[item] = value
200
201     def __delitem__(self, item):
202         del self.__str__[item]
203
204     def __len__(self):
205         return len(self.__str__)
206
207     def __iter__(self):
208         return iter(self.__str__)
209
210     def __contains__(self, item):
211         return item in self.__str__
212
213     def __getitem__(self, item):
214         return self.__str__[item]
215
216     def __setitem__(self, item, value):
217         self.__str__[item] = value
218
219     def __delitem__(self, item):
220         del self.__str__[item]
221
222     def __len__(self):
223         return len(self.__str__)
224
225     def __iter__(self):
226         return iter(self.__str__)
227
228     def __contains__(self, item):
229         return item in self.__str__
230
231     def __getitem__(self, item):
232         return self.__str__[item]
233
234     def __setitem__(self, item, value):
235         self.__str__[item] = value
236
237     def __delitem__(self, item):
238         del self.__str__[item]
239
240     def __len__(self):
241         return len(self.__str__)
242
243     def __iter__(self):
244         return iter(self.__str__)
245
246     def __contains__(self, item):
247         return item in self.__str__
248
249     def __getitem__(self, item):
250         return self.__str__[item]
251
252     def __setitem__(self, item, value):
253         self.__str__[item] = value
254
255     def __delitem__(self, item):
256         del self.__str__[item]
257
258     def __len__(self):
259         return len(self.__str__)
260
261     def __iter__(self):
262         return iter(self.__str__)
263
264     def __contains__(self, item):
265         return item in self.__str__
266
267     def __getitem__(self, item):
268         return self.__str__[item]
269
270     def __setitem__(self, item, value):
271         self.__str__[item] = value
272
273     def __delitem__(self, item):
274         del self.__str__[item]
275
276     def __len__(self):
277         return len(self.__str__)
278
279     def __iter__(self):
280         return iter(self.__str__)
281
282     def __contains__(self, item):
283         return item in self.__str__
284
285     def __getitem__(self, item):
286         return self.__str__[item]
287
288     def __setitem__(self, item, value):
289         self.__str__[item] = value
290
291     def __delitem__(self, item):
292         del self.__str__[item]
293
294     def __len__(self):
295         return len(self.__str__)
296
297     def __iter__(self):
298         return iter(self.__str__)
299
300     def __contains__(self, item):
301         return item in self.__str__
302
303     def __getitem__(self, item):
304         return self.__str__[item]
305
306     def __setitem__(self, item, value):
307         self.__str__[item] = value
308
309     def __delitem__(self, item):
310         del self.__str__[item]
311
312     def __len__(self):
313         return len(self.__str__)
314
315     def __iter__(self):
316         return iter(self.__str__)
317
318     def __contains__(self, item):
319         return item in self.__str__
320
321     def __getitem__(self, item):
322         return self.__str__[item]
323
324     def __setitem__(self, item, value):
325         self.__str__[item] = value
326
327     def __delitem__(self, item):
328         del self.__str__[item]
329
330     def __len__(self):
331         return len(self.__str__)
332
333     def __iter__(self):
334         return iter(self.__str__)
335
336     def __contains__(self, item):
337         return item in self.__str__
338
339     def __getitem__(self, item):
340         return self.__str__[item]
341
342     def __setitem__(self, item, value):
343         self.__str__[item] = value
344
345     def __delitem__(self, item):
346         del self.__str__[item]
347
348     def __len__(self):
349         return len(self.__str__)
350
351     def __iter__(self):
352         return iter(self.__str__)
353
354     def __contains__(self, item):
355         return item in self.__str__
356
357     def __getitem__(self, item):
358         return self.__str__[item]
359
360     def __setitem__(self, item, value):
361         self.__str__[item] = value
362
363     def __delitem__(self, item):
364         del self.__str__[item]
365
366     def __len__(self):
367         return len(self.__str__)
368
369     def __iter__(self):
370         return iter(self.__str__)
371
372     def __contains__(self, item):
373         return item in self.__str__
374
375     def __getitem__(self, item):
376         return self.__str__[item]
377
378     def __setitem__(self, item, value):
379         self.__str__[item] = value
380
381     def __delitem__(self, item):
382         del self.__str__[item]
383
384     def __len__(self):
385         return len(self.__str__)
386
387     def __iter__(self):
388         return iter(self.__str__)
389
390     def __contains__(self, item):
391         return item in self.__str__
392
393     def __getitem__(self, item):
394         return self.__str__[item]
395
396     def __setitem__(self, item, value):
397         self.__str__[item] = value
398
399     def __delitem__(self, item):
400         del self.__str__[item]
401
402     def __len__(self):
403         return len(self.__str__)
404
405     def __iter__(self):
406         return iter(self.__str__)
407
408     def __contains__(self, item):
409         return item in self.__str__
410
411     def __getitem__(self, item):
412         return self.__str__[item]
413
414     def __setitem__(self, item, value):
415         self.__str__[item] = value
416
417     def __delitem__(self, item):
418         del self.__str__[item]
419
420     def __len__(self):
421         return len(self.__str__)
422
423     def __iter__(self):
424         return iter(self.__str__)
425
426     def __contains__(self, item):
427         return item in self.__str__
428
429     def __getitem__(self, item):
430         return self.__str__[item]
431
432     def __setitem__(self, item, value):
433         self.__str__[item] = value
434
435     def __delitem__(self, item):
436         del self.__str__[item]
437
438     def __len__(self):
439         return len(self.__str__)
440
441     def __iter__(self):
442         return iter(self.__str__)
443
444     def __contains__(self, item):
445         return item in self.__str__
446
447     def __getitem__(self, item):
448         return self.__str__[item]
449
450     def __setitem__(self, item, value):
451         self.__str__[item] = value
452
453     def __delitem__(self, item):
454         del self.__str__[item]
455
456     def __len__(self):
457         return len(self.__str__)
458
459     def __iter__(self):
460         return iter(self.__str__)
461
462     def __contains__(self, item):
463         return item in self.__str__
464
465     def __getitem__(self, item):
466         return self.__str__[item]
467
468     def __setitem__(self, item, value):
469         self.__str__[item] = value
470
471     def __delitem__(self, item):
472         del self.__str__[item]
473
474     def __len__(self):
475         return len(self.__str__)
476
477     def __iter__(self):
478         return iter(self.__str__)
479
480     def __contains__(self, item):
481         return item in self.__str__
482
483     def __getitem__(self, item):
484         return self.__str__[item]
485
486     def __setitem__(self, item, value):
487         self.__str__[item] = value
488
489     def __delitem__(self, item):
490         del self.__str__[item]
491
492     def __len__(self):
493         return len(self.__str__)
494
495     def __iter__(self):
496         return iter(self.__str__)
497
498     def __contains__(self, item):
499         return item in self.__str__
500
501     def __getitem__(self, item):
502         return self.__str__[item]
503
504     def __setitem__(self, item, value):
505         self.__str__[item] = value
506
507     def __delitem__(self, item):
508         del self.__str__[item]
509
510     def __len__(self):
511         return len(self.__str__)
512
513     def __iter__(self):
514         return iter(self.__str__)
515
516     def __contains__(self, item):
517         return item in self.__str__
518
519     def __getitem__(self, item):
520         return self.__str__[item]
521
522     def __setitem__(self, item, value):
523         self.__str__[item] = value
524
525     def __delitem__(self, item):
526         del self.__str__[item]
527
528     def __len__(self):
529         return len(self.__str__)
530
531     def __iter__(self):
532         return iter(self.__str__)
533
534     def __contains__(self, item):
535         return item in self.__str__
536
537     def __getitem__(self, item):
538         return self.__str__[item]
539
540     def __setitem__(self, item, value):
541         self.__str__[item] = value
542
543     def __delitem__(self, item):
544         del self.__str__[item]
545
546     def __len__(self):
547         return len(self.__str__)
548
549     def __iter__(self):
550         return iter(self.__str__)
551
552     def __contains__(self, item):
553         return item in self.__str__
554
555     def __getitem__(self, item):
556         return self.__str__[item]
557
558     def __setitem__(self, item, value):
559         self.__str__[item] = value
560
561     def __delitem__(self, item):
562         del self.__str__[item]
563
564     def __len__(self):
565         return len(self.__str__)
566
567     def __iter__(self):
568         return iter(self.__str__)
569
570     def __contains__(self, item):
571         return item in self.__str__
572
573     def __getitem__(self, item):
574         return self.__str__[item]
575
576     def __setitem__(self, item, value):
577         self.__str__[item] = value
578
579     def __delitem__(self, item):
580         del self.__str__[item]
581
582     def __len__(self):
583         return len(self.__str__)
584
585     def __iter__(self):
586         return iter(self.__str__)
587
588     def __contains__(self, item):
589         return item in self.__str__
590
591     def __getitem__(self, item):
592         return self.__str__[item]
593
594     def __setitem__(self, item, value):
595         self.__str__[item] = value
596
597     def __delitem__(self, item):
598         del self.__str__[item]
599
600     def __len__(self):
601         return len(self.__str__)
602
603     def __iter__(self):
604         return iter(self.__str__)
605
606     def __contains__(self, item):
607         return item in self.__str__
608
609     def __getitem__(self, item):
610         return self.__str__[item]
611
612     def __setitem__(self, item, value):
613         self.__str__[item] = value
614
615     def __delitem__(self, item):
616         del self.__str__[item]
617
618     def __len__(self):
619         return len(self.__str__)
620
621     def __iter__(self):
622         return iter(self.__str__)
623
624     def __contains__(self, item):
625         return item in self.__str__
626
627     def __getitem__(self, item):
628         return self.__str__[item]
629
630     def __setitem__(self, item, value):
631         self.__str__[item] = value
632
633     def __delitem__(self, item):
634         del self.__str__[item]
635
636     def __len__(self):
637         return len(self.__str__)
638
639     def __iter__(self):
640         return iter(self.__str__)
641
642     def __contains__(self, item):
643         return item in self.__str__
644
645     def __getitem__(self, item):
646         return self.__str__[item]
647
648     def __setitem__(self, item, value):
649         self.__str__[item] = value
650
651     def __delitem__(self, item):
652         del self.__str__[item]
653
654     def __len__(self):
655         return len(self.__str__)
656
657     def __iter__(self):
658         return iter(self.__str__)
659
660     def __contains__(self, item):
661         return item in self.__str__
662
663     def __getitem__(self, item):
664         return self.__str__[item]
665
666     def __setitem__(self, item, value):
667         self.__str__[item] = value
668
669     def __delitem__(self, item):
670         del self.__str__[item]
671
672     def __len__(self):
673         return len(self.__str__)
674
675     def __iter__(self):
676         return iter(self.__str__)
677
678     def __contains__(self, item):
679         return item in self.__str__
680
681     def __getitem__(self, item):
682         return self.__str__[item]
683
684     def __setitem__(self, item, value):
685         self.__str__[item] = value
686
687     def __delitem__(self, item):
688         del self.__str__[item]
689
690     def __len__(self):
691         return len(self.__str__)
692
693     def __iter__(self):
694         return iter(self.__str__)
695
696     def __contains__(self, item):
697         return item in self.__str__
698
699     def __getitem__(self, item):
700         return self.__str__[item]
701
702     def __setitem__(self, item, value):
703         self.__str__[item] = value
704
705     def __delitem__(self, item):
706         del self.__str__[item]
707
708     def __len__(self):
709         return len(self.__str__)
710
711     def __iter__(self):
712         return iter(self.__str__)
713
714     def __contains__(self, item):
715         return item in self.__str__
716
717     def __getitem__(self, item):
718         return self.__str__[item]
719
720     def __setitem__(self, item, value):
721         self.__str__[item] = value
722
723     def __delitem__(self, item):
724         del self.__str__[item]
725
726     def __len__(self):
727         return len(self.__str__)
728
729     def __iter__(self):
730         return iter(self.__str__)
731
732     def __contains__(self, item):
733         return item in self.__str__
734
735     def __getitem__(self, item):
736         return self.__str__[item]
737
738     def __setitem__(self, item, value):
739         self.__str__[item] = value
740
741     def __delitem__(self, item):
742         del self.__str__[item]
743
744     def __len__(self):
745         return len(self.__str__)
746
747     def __iter__(self):
748         return iter(self.__str__)
749
750     def __contains__(self, item):
751         return item in self.__str__
752
753     def __getitem__(self, item):
754         return self.__str__[item]
755
756     def __setitem__(self, item, value):
757         self.__str__[item] = value
758
759     def __delitem__(self, item):
760         del self.__str__[item]
761
762     def __len__(self):
763         return len(self.__str__)
764
765     def __iter__(self):
766         return iter(self.__str__)
767
768     def __contains__(self, item):
769         return item in self.__str__
770
771     def __getitem__(self, item):
772         return self.__str__[item]
773
774     def __setitem__(self, item, value):
775         self.__str__[item] = value
776
777     def __delitem__(self, item):
778         del self.__str__[item]
779
780     def __len__(self):
781         return len(self.__str__)
782
783     def __iter__(self):
784         return iter(self.__str__)
785
786     def __contains__(self, item):
787         return item in self.__str__
788
789     def __getitem__(self, item):
790         return self.__str__[item]
791
792     def __setitem__(self, item, value):
793         self.__str__[item] = value
794
795     def __delitem__(self, item):
796         del self.__str__[item]
797
798     def __len__(self):
799         return len(self.__str__)
800
801     def __iter__(self):
802         return iter(self.__str__)
803
804     def __contains__(self, item):
805         return item in self.__str__
806
807     def __getitem__(self, item):
808         return self.__str__[item]
809
810     def __setitem__(self, item, value):
811         self.__str__[item] = value
812
813     def __delitem__(self, item):
814         del self.__str__[item]
815
816     def __len__(self):
817         return len(self.__str__)
818
819     def __iter__(self):
820         return iter(self.__str__)
821
822     def __contains__(self, item):
823         return item in self.__str__
824
825     def __getitem__(self, item):
826         return self.__str__[item]
827
828     def __setitem__(self, item, value):
829         self.__str__[item] = value
830
831     def __delitem__(self, item):
832         del self.__str__[item]
833
834     def __len__(self):
835         return len(self.__str__)
836
837     def __iter__(self):
838         return iter(self.__str__)
839
840     def __contains__(self, item):
841         return item in self.__str__
842
843     def __getitem__(self, item):
844         return self.__str__[item]
845
846     def __setitem__(self, item, value):
847         self.__str__[item] = value
848
849     def __delitem__(self, item):
850         del self.__str__[item]
851
852     def __len__(self):
853         return len(self.__str__)
854
855     def __iter__(self):
856         return iter(self.__str__)
857
858     def __contains__(self, item):
859         return item in self.__str__
860
861     def __getitem__(self, item):
862         return self.__str__[item]
863
864     def __setitem__(self, item, value):
865         self.__str__[item] = value
866
867     def __delitem__(self, item):
868         del self.__str__[item]
869
870     def __len__(self):
871         return len(self.__str__)
872
873     def __iter__(self):
874         return iter(self.__str__)
875
876     def __contains__(self, item):
877         return item in self.__str__
878
879     def __getitem__(self, item):
880         return self.__str__[item]
881
882     def __setitem__(self, item, value):
883         self.__str__[item] = value
884
885     def __delitem__(self, item):
886         del self.__str__[item]
887
888     def __len__(self):
889         return len(self.__str__)
890
891     def __iter__(self):
892         return iter(self.__str__)
893
894     def __contains__(self, item):
895         return item in self.__str__
896
897     def __getitem__(self, item):
898         return self.__str__[item]
899
900     def __setitem__(self, item, value):
901         self.__str__[item] = value
902
903     def __delitem__(self, item):
904         del self.__str__[item]
905
906     def __len__(self):
907         return len(self.__str__)
908
909     def __iter__(self):
910         return iter(self.__str__)
911
912     def __contains__(self, item):
913         return item in self.__str__
914
915     def __getitem__(self, item):
916         return self.__str__[item]
917
918     def __setitem__(self, item, value):
919         self.__str__[item] = value
920
921     def __delitem__(self, item):
922         del self.__str__[item]
923
924     def __len__(self):
925         return len(self.__str__)
926
927     def __iter__(self):
928         return iter(self.__str__)
929
930     def __contains__(self, item):
931         return item in self.__str__
932
933     def __getitem__(self, item):
934         return self.__str__[item]
935
936     def __setitem__(self, item, value):
937         self.__str__[item] = value
938
939     def __delitem__(self, item):
940         del self.__str__[item]
941
942     def __len__(self):
943         return len(self.__str__)
944
945     def __iter__(self):
946         return iter(self.__str__)
947
948     def __contains__(self, item):
949         return item in self.__str__
950
951     def __getitem__(self, item):
952         return self.__str__[item]
953
954     def __setitem__(self, item, value):
955         self.__str__[item] = value
956
957     def __delitem__(self, item):
958         del self.__str__[item]
959
960     def __len__(self):
961         return len(self.__str__)
962
963     def __iter__(self):
964         return iter(self.__str__)
965
966     def __contains__(self, item):
967         return item in self.__str__
968
969     def __getitem__(self, item):
970         return self.__str__[item]
971
972     def __setitem__(self, item, value):
973         self.__str__[item] = value
974
975     def __delitem__(self, item):
976         del self.__str__[item]
977
978     def __len__(self):
979         return len(self.__str__)
980
981     def __iter__(self):
982         return iter(self.__str__)
983
984     def __contains__(self, item):
985         return item in self.__str__
986
987     def __getitem__(self, item):
988         return self.__str__[item]
989
990     def __setitem__(self, item, value):
991         self.__str__[item] = value
992
993     def __delitem__(self, item):
994         del self.__str__[item]
995
996     def __len__(self):
997         return len(self.__str__)
1000
1001     def __iter__(self):
1002         return iter(self.__str__)
1003
1004     def __contains__(self, item):
1005         return item in self.__str__
1006
1007     def __getitem__(self, item):
1008         return self.__str__[item]
1009
1010     def __setitem__(self, item, value):
1011         self.__str__[item] = value
1012
1013     def __delitem__(self, item):
1014         del self.__str__[item]
1015
1016     def __len__(self):
1017         return len(self.__str__)
1018
1019     def __iter__(self):
1020         return iter(self.__str__)
1021
1022     def __contains__(self, item):
1023         return item in self.__str__
1024
1025     def __getitem__(self, item):
1026         return self.__str__[item]
1027
1028     def __setitem__(self, item, value):
1029         self.__str__[item] = value
1030
1031     def __delitem__(self, item):
1032         del self.__str__[item]
1033
1034     def __len__(self):
1035         return len(self.__str__)
1036
1037     def __iter__(self):
1038         return iter(self.__str__)
1039
1040     def __contains__(self, item):
1041         return item in self.__str__
1042
1043     def __getitem__(self, item):
1044         return self.__str__[item]
1045
1046     def __setitem__(self, item, value):
1047         self.__str__[item] = value
1048
1049     def __delitem__(self, item):
1050         del self.__str__[item]
1051
1052     def __len__(self):
1053         return len(self.__str__)
1054
1055     def __iter__(self):
1056         return iter(self.__str__)
1057
1058     def __contains__(self, item):
1059         return item in self.__str__
1060
1061     def __getitem__(self, item):
1062         return self.__str__[item]
1063
1064     def __setitem__(self, item, value):
1065         self.__str__[item] = value
1066
1067     def __delitem__(self, item):
1068         del self.__str__[item]
1069
1070     def __len__(self):
1071         return len(self.__str__)
1072
1073     def __iter__(self):
1074         return iter(self.__str__)
1075
1076     def __contains__(self, item):
1077         return item in self.__str__
1078
1079     def __getitem__(self, item):
1080         return self.__str__[item]
1081
1082     def __setitem__(self, item, value):
1083         self.__str__[item] = value
1084
1085     def __delitem__(self, item):
1086         del self.__str__[item]
1087
1088     def __len__(self):
1089         return len(self.__str__)
1090
1091     def __iter__(self):
1092         return iter(self.__str__)
1093
1094     def __contains__(self, item):
1095         return item in self.__str__
1096
1097     def __getitem__(self, item):
1098         return self.__str__[item]
1099
1100     def __setitem__(self, item, value):
1101         self.__str__[item] = value
1102
1103     def __delitem__(self, item):
1104         del self.__str__[item]
1105
1106     def __len__(self):
1107         return len(self.__str__)
1108
1109     def __iter__(self):
1110         return iter(self.__str__)
1111
1112     def __contains__(self, item):
1113         return item in self.__str__
1114
1115     def __getitem__(self, item):
1116         return self.__str__[item]
1117
1118     def __setitem__(self, item, value):
1119         self.__str__[item] = value
1120
1121     def __delitem__(self, item):
1122         del self.__str__[item]
1123
1124     def __len__(self):
1125         return len(self.__str__)
1126
1127     def __iter__(self):
1128         return iter(self.__str__)
1129
1130     def __contains__(self, item):
1131         return item in self.__str__
1132
1133     def __getitem__(self, item):
1134         return self.__str__[item]
1135
1136     def __setitem__(self, item, value):
1137         self.__str__[item] = value
1138
1139     def __delitem__(self, item):
1140         del self.__str__[item]
1141
1142     def __len__(self):
1143         return len(self.__str__)
1144
1145     def __iter__(self):
1146         return iter(self.__str__)
1147
1148     def __contains__(self, item):
1149         return item in self.__str__
1150
1151     def __getitem__(self, item):
1152         return self.__str__[item]
1153
1154     def __setitem__(self, item, value):
1155         self.__str__[item] = value
1156
1157     def __delitem__(self, item):
1158         del self.__str__[item]
1159
1160     def __len__(self):
1161         return len(self.__str__)
1162
1163     def __iter__(self):
1164         return iter(self.__str__)
1165
1166     def __contains__(self, item):
1167         return item in self.__str__
1168
1169     def __getitem__(self, item):
1170         return self.__str__[item]
1171
1172     def __setitem__(self, item, value):
1173         self.__str__[item] = value
1174
1175     def __delitem__(self, item):
1176         del self.__str__[item]
1177
1178     def __len__(self):
117
```

```
master.py - webots_ros2_tutorials - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
  OPEN EDITORS
    webots_ros2_tutorials
  WEBOTS_ROS2_TUTORIALS
    launch
    proto
    Robot_sensor_proto
    resource
    test
    webots_ros2_tutorials
    _init_.py
    master.py
    move.py
    worlds
    textures
    custom_line_follower.world
    custom_line_follower.wbt
    robot.wbt
    package.xml
    setup.py
    setup.py

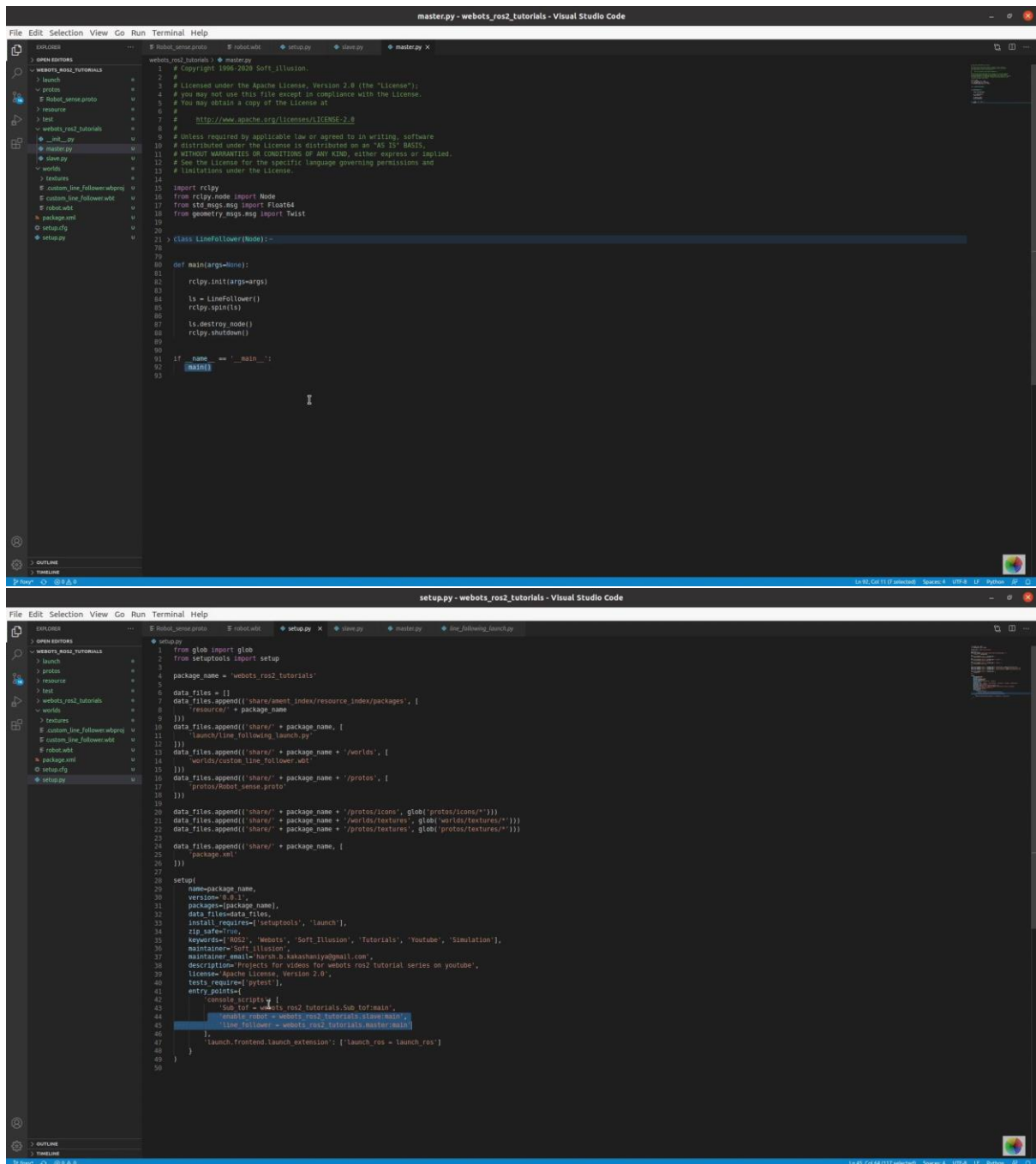
1 # Copyright 1996-2020 Soft Illusion.
2 #
3 # Licensed under the Apache License, Version 2.0 (the "License");
4 # you may not use this file except in compliance with the License.
5 # You may obtain a copy of the License at
6 #
7 # http://www.apache.org/licenses/LICENSE-2.0
8 #
9 # Unless required by 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 rclpy
16 from rclpy.node import Node
17 from std_msgs.msg import Float64
18 from geometry_msgs.msg import Twist
19
20 class LineFollower(Node):
21
22     def __init__(self):
23         super().__init__('linefollower_cmdvel')
24         # Subscribe to sensors
25         self.subs_right_ir = self.create_subscription(Float64, 'right_IR', self.rightIR_cb, 1)
26         self.subs_left_ir = self.create_subscription(Float64, 'left_IR', self.leftIR_cb, 1)
27         self.subs_mid_ir = self.create_subscription(Float64, 'mid_IR', self.midIR_cb, 1)
28         # Publish cmd vel
29         self.pub_cmdvel = self.create_publisher(Twist, 'cmd_vel', 1)
30
31         # vehicle parameters
32         self.speed = 0.2
33         self.angle_correction = 0.01
34
35         # Initialize parameters
36         self.GS_PIDOF, self.GS_MD, self.GS_LEFT = 0, 0, 0
37         self.deltaS = 0
38         self.cmd = Twist()
39         self.stop = False
40         self.count = 0
41         self.count_threshold = 10
42
43     def LineFollowingModule(self):
44         # Call backs to update sensor reading variables
45         def rightIR_cb(self, msg):
46             pass
47         def leftIR_cb(self, msg):
48             pass
49         def midIR_cb(self, msg):
50             pass
51
52     def mainargs=None):
53
54 if __name__ == '__main__':
55     main()
56
```



## Tahap 4

Setelah melihat dan mengatur review dan mengatur master project, kita mulai melakukan setup pada project.





## Tahap 5

Setelah mengatur dan setup project, kita melanjutkan tahap terakhir untuk build dan mendemonstrasikan robot yang sudah kita buat.





