A logo with black and red lines

Description automatically generated**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

PROJECT REPORT

Distributed Control Systems

Authors: **Dafinoiu Andreea-Raluca (30342)**

**Apetrei Andrei-Octavian (30343)**

Instructor: **As. Dr. Ing. Dahlia AL-JANABI**

2024

1. **Introduction:**

This project purposes to model and control two intersections with the use of OETPN which communicate via network.

The intersections modelled in this project are:

* [https://www.google.com/maps/place/47°04'15.9"N+21°55'12.2"E/@47.071094,21.920067,17z/data=!3m1!4b1!4m4!3m3!8m2!3d47.071094!4d21.920067?entry=ttu](https://www.google.com/maps/place/47°04'15.9%22N+21°55'12.2%22E/@47.071094,21.920067,17z/data=!3m1!4b1!4m4!3m3!8m2!3d47.071094!4d21.920067?entry=ttu)
* <https://www.google.com/maps/search/47.069259,+21.924812?entry=tts>

An aerial view of a road intersection

Description automatically generated

Figure : First Intersection

An aerial view of a road intersection

Description automatically generated

Figure : Second Intersection

****

Figure : Both intersection with the middle street

1. **Petri Nets:**

The Petri Nets for both intersection, middle street and the controller are presented bellow.

1. First Intersection:

A diagram of a machine

Description automatically generated

Figure : Petri Net of the First Intersection

1. Second Intersection:

A diagram of a network

Description automatically generated

A diagram of a network

Description automatically generated

A diagram of a flowchart

Description automatically generated

1. Middle Street:

A diagram of a diagram

Description automatically generated

1. Controller:

A diagram of a network

Description automatically generated

1. **Guards & maps:**
2. **First Intersection:**

**Place types:**

* **DataCar:** i1, e1, i2, e2, i3, e3, oli1, ole1, oli2, ole2, oli3, ole3
* **DataCarQueue:** ili1, ile1, ili2, ile2, ili3, ile3
* **DataTransfer:** P\_oli2

**Guard & Map:**

* **Ti11**

(i1 != null) & (ili1.CanAddCars)

ili1.AddElement(i1)

* **Ti12**

(ili1.HaveCarForMe) & (ili2.CanAddCars)

Ili2.PopElementWithTargetToQueue(ili1)

* **Ti13**

(ili2.HaveCarForMe) & (ili3.CanAddCars)

Ili3. PopElementWithTargetToQueue (ili2)

* **Ti14**

(ili3.HaveCarForMe)

Ili3.PopElementWithTarget(oli1)

* **Ti15**

(ili2.HaveCarForMe)

Ili2.PopElementWithTarget(oli3)

* **Ti16**

(ili3.HaveCarForMe) & (ili1.CanAddCars)

Ili1. PopElementWithTargetToQueue (ili3)

* **Ti17**

(ili1.HaveCarForMe)

Ili1.PopElementWithTarget(oli2)

* **Te11**

(e1 != null) & (ile1.CanAddCars)

Ile1.AddElement(e1)

* **Te12**

(ile1.HaveCarForMe) & (ile2.CanAddCars)

Ile2. PopElementWithTargetToQueue (ile1)

* **Te13**

(ile2.HaveCarForMe)

Ile2.PopElementWithTarget(ole3)

* **Te14**

(ile2.HaveCarForMe) & (ile3.CanAddCars)

Ile3. PopElementWithTargetToQueue (ile2)

* **Te15**

(ile3.HaveCarForMe)

Ile3.PopElementWithTarget(ole1)

* **Te16**

(ile3.HaveCarForMe) & (ile1.CanAddCars)

Ile1. PopElementWithTargetToQueue (ile3)

* **Te17**

(ile1.HaveCarForMe)

Ile1.PopElementWithTarget(ole2)

* **Ti21**

(i2 != null) & (ili2.CanAddCars)

Ili2.AddElement(i2)

* **Ti22**

(ili2.HaveCarForMe)

Ili2.PopElementWithTarget(oli3)

* **Te21**

(e2 != null) & (ile2.CanAddCars)

Ile2.AddElement(e2)

* **Te22**

(ile2.HaveCarForMe)

Ile2.PopElementWithTarget(ole3)

* **Ti31**

(i3 != null) & (ili3.CanAddCars)

Ili3.AddElement(i3)

* **Te31**

(e3 != null) & (ile3.CanAddCars)

Ile3.AddElement(i3)

1. **Second Intersection:**

**Place Types:**

* **DataCar:**
  + P\_ai1, P\_bi1, P\_am1, P\_bm1, P\_ae1, P\_be1
  + P\_ai2, P\_am2, P\_ae2
  + P\_ai3, P\_am3, P\_ae3
  + P\_ai4, P\_am4, P\_ae4
  + oli1, olm1, ole1
  + oli2, olm2, ole2
  + oli3, olm3, ole3
  + oli4, olm4, ole4
  + P\_ci1
  + P\_di1, P\_dm1, P\_de1
  + P\_ei1, P\_em1, P\_ee1
* **DataCarQueue:**
  + ili1, ilm1, ile1
  + ili2, ilm2, ile2
  + ili3, ilm3, ile3
  + ili4, ilm4, ile4
* **DataString:** green, full, P\_TL1, P\_TL2, P\_TL3, P\_TL4
* **DataTransfer:** OP1, OP2, OP3, OP4, E\_oli3

**Guards & maps:**

* **T\_ai1**

(P\_ai1 != null) & (ili1.CanAddCars)

Ili1.AddElement(P\_ai1)

(P\_ai1 != null) & (ili1.CanNotAddCars)

OP1.SendOverNetwork(full)

P\_ai1.Copy(P\_ai1)

* **T\_bi1**

(P\_TL1 == green) & (ili1.HaveCar)

P\_bi1.PopElementWithoutTarget(ili1)

P\_TL1.Move(P\_TL1)

* **T\_ai2**

(P\_ai2 != null) & (ili2.CanAddCars)

Ili2.AddElement(P\_ai2)

(P\_ai2 != null) & (ili2.CanNotAddCars)

OP2.SendOverNetwork(full)

P\_ai2.Copy(P\_ai2)

* **T\_ai3**

(P\_ai3 != null) & (ili3.CanAddCars)

Ili3.AddElement(P\_ai3)

(P\_ai3 != null) & (ili3.CanNotAddCars)

OP3.SendOverNetwork(full)

P\_ai3.Copy(P\_ai3)

* **T\_ai4**

(P\_ai4 != null) & (ili4.CanAddCars)

Ili4.AddElement(P\_ai4)

(P\_ai4 != null) & (ili4.CanNotAddCars)

OP4.SendOverNetwork(full)

P\_ai4.Copy(P\_ai4)

* **T\_am1**

(P\_am1 != null) & (ilm1.CanAddCars)

Ilm1.AddElement(P\_am1)

(P\_am1 != null) & (ilm1.CanNotAddCars)

OP1.SendOverNetwork(full)

P\_am1.Copy(P\_am1)

* **T\_bm1**

(P\_TL1 == green) & (ilm1.HaveCar)

P\_bm1.PopElementWithoutTarget(ilm1)

P\_TL1.Move(P\_TL1)

* **T\_am2**

(P\_am2 != null) & (ilm2.CanAddCars)

Ilm2.AddElement(P\_am2)

(P\_am2 != null) & (ilm2.CanNotAddCars)

OP2.SendOverNetwork(full)

P\_am2.Copy(P\_am2)

* **T\_bm2**

(P\_TL2 == green) & (ilm2.HaveCar)

P\_cm1.PopElementWithoutTarget(ilm2)

P\_TL2.Move(P\_TL2)

* **T\_am3**

(P\_am3 != null) & (ilm3.CanAddCars)

Ilm3.AddElement(P\_am3)

(P\_am3 != null) & (ilm3.CanNotAddCars)

OP3.SendOverNetwork(full)

P\_am3.Copy(P\_am3)

* **T\_am4**

(P\_am4 != null) & (ilm4.CanAddCars)

Ilm4.AddElement(P\_am4)

(P\_am4 != null) & (ilm4.CanNotAddCars)

OP4.SendOverNetwork(full)

P\_am4.Copy(P\_am4)

* **T\_ae1**

(P\_ae1 != null) & (ile1.CanAddCars)

Ile1.AddElement(P\_ae1)

(P\_ae1 != null) & (ile1.CanNotAddCars)

OP1.SendOverNetwork(full)

P\_ae1.Copy(P\_ae1)

* **T\_be1**

(P\_TL1 == green) & (ile1.HaveCar)

P\_be1.PopElementWithoutTarget(ile1)

P\_TL1.Move(P\_TL1)

* **T\_ae2**

(P\_ae2 != null) & (ile2.CanAddCars)

Ile2.AddElement(P\_ae2)

(P\_ae2 != null) & (ile2.CanNotAddCars)

OP2.SendOverNetwork(full)

P\_ae2.Copy(P\_ae2)

* **T\_be2**

(P\_TL2 == green) & (ile2.HaveCar)

P\_ce1.PopElementWithoutTarget(ile2)

P\_TL2.Move(P\_TL2)

* **T\_ae3**

(P\_ae3 != null) & (ile3.CanAddCars)

Ile3.AddElement(P\_ae3)

(P\_ae3 != null) & (ile3.CanNotAddCars)

OP3.SendOverNetwork(full)

P\_ae3.Copy(P\_ae3)

* **T\_ae4**

(P\_ae4 != null) & (ile4.CanAddCars)

Ile4.AddElement(P\_ae3)

(P\_ae4 != null) & (ile4.CanNotAddCars)

OP4.SendOverNetwork(full)

P\_ae4.Copy(P\_ae4)

* **T\_ai12:**

(P\_bi1 != null) & (ili2.CanAddCars)

Ile2.AddElement(P\_ bi1)

(P\_ bi1!= null) & (ili2.CanNotAddCars)

OP2.SendOverNetwork(full)

P\_ bi1.Copy(P\_ bi1)

* **T\_bi12:**

(P\_TL2 == green) & (ili2. HaveCarForMe)

P\_ci1.PopElementWithoutTarget(ili2)

P\_TL2.Move(P\_TL2)

* **T\_ai23**

(P\_ci1 != null) & (ili3.CanAddCars)

Ili3.AddElement(P\_ ci1)

(P\_ ci1!= null) & (ili3.CanNotAddCars)

OP3.SendOverNetwork(full)

P\_ ci1.Copy(P\_ ci1)

* **T\_bi23**

(P\_TL3 == green) & (ili3. HaveCarForMe)

P\_di1.PopElementWithoutTarget(ili3)

P\_TL3.Move(P\_TL3)

* **T\_di14**

(P\_di1 != null) & (ili4.CanAddCars)

Ili4.AddElement(P\_ di1)

(P\_ di1!= null) & (ili4.CanNotAddCars)

OP4.SendOverNetwork(full)

P\_ di1.Copy(P\_ di1)

* **T\_ei14**

(P\_TL4 == green) & (ili4. HaveCarForMe)

P\_ei1.PopElementWithoutTarget(ili3)

P\_TL4.Move(P\_TL4)

* **T\_ei1**

(P\_ei1 != null) & (ili1.CanAddCars)

Ili1.AddElement(P\_ ei1)

(P\_ ei1!= null) & (ili1.CanNotAddCars)

OP3.SendOverNetwork(full)

P\_ ei1.Copy(P\_ ei1)

* **T\_am12**

(P\_bm1 != null) & (ilm2.CanAddCars)

Ilm2.AddElement(P\_ bm1)

(P\_ bm1!= null) & (ilm2.CanNotAddCars)

OP2.SendOverNetwork(full)

P\_ bm1.Copy(P\_ bm1)

* **T\_bm23**

(P\_TL3 == green) & (ilm3.HaveCarForMe)

P\_dm1.PopElementWithoutTarget(ilm3)

P\_TL3.Move(P\_TL3)

* **T\_dm14**

(P\_dm1 != null) & (ilm4.CanAddCars)

Ilm4.AddElement(P\_ dm1)

(P\_ dm1!= null) & (ilm4.CanNotAddCars)

OP4.SendOverNetwork(full)

P\_ dm1.Copy(P\_ dm1)

* **T\_em14**

(P\_TL4 == green) & (ilm4.HaveCarForMe)

P\_em1.PopElementWithoutTarget(ilm4)

P\_TL4.Move(P\_TL4)

* **T\_ae12**

(P\_be1 != null) & (ile2.CanAddCars)

Ile2.AddElement(P\_ be 1)

(P\_ be 1!= null) & (ile2.CanNotAddCars)

OP2.SendOverNetwork(full)

P\_ be 1.Copy(P\_ be 1)

* **T\_be23**

(P\_TL3 == green) & (ile3.HaveCarForMe)

P\_de1.PopElementWithoutTarget(ile3)

P\_TL3.Move(P\_TL3)

* **T\_de14**

(P\_de1 != null) & (ile4.CanAddCars)

Ile4.AddElement(P\_de1)

(P\_ de1!= null) & (ile4.CanNotAddCars)

OP4.SendOverNetwork(full)

P\_de1.Copy(P\_de1)

* **T\_ee14**

(P\_TL4 == green) & (ile4.HaveCarForMe)

P\_ee1.PopElementWithoutTarget(ile4)

P\_TL4.Move(P\_TL4)

* **Toe42**

(P\_TL1 == green) & (ile1.HaveCarForMe)

Ole1.PopElementWithoutTarget(ile1)

P\_TL1.Move(P\_TL1)

* **Toe13**

(P\_TL1 == green) & (ile1.HaveCarForMe)

Ole2.PopElementWithoutTarget(ile1)

P\_TL1.Move(P\_TL1)

* **Toi23**

(P\_TL2 == green) & (ili2.HaveCarForMe)

Oli3.PopElementWithoutTarget(ili2)

P\_TL2.Move(P\_TL2)

* **Tom23**

(P\_TL2 == green) & (ilm2.HaveCarForMe)

Olm3.PopElementWithoutTarget(ilm2)

P\_TL2.Move(P\_TL2)

* **Toe22**

(P\_TL2 == green) & (ile2.HaveCarForMe)

Oli2.PopElementWithoutTarget(ile2)

P\_TL2.Move(P\_TL2)

* **Toe33**

(P\_TL3 == green) & (ile3.HaveCar)

Ole2.PopElementWithTarget(ile3)

P\_TL3.Move(P\_TL3)

* **Toe44**

(P\_TL4 == green) & (ile4.HaveCarForMe)

Ole4.PopElementWithoutTarget(ile4)

P\_TL4.Move(P\_TL4)

* **Toi34**

(P\_TL3 == green) & (ili3.HaveCarForMe)

Oli4.PopElementWithoutTarget(ili3)

P\_TL3.Move(P\_TL3)

* **Tom34**

(P\_TL3 == green) & (ilm3.HaveCarForMe)

Oli4.PopElementWithoutTarget(ilm3)

P\_TL3.Move(P\_TL3)

* **Toe34**

(P\_TL3 == green) & (ile3.HaveCarForMe)

Ole4.PopElementWithoutTarget(ile3)

P\_TL3.Move(P\_TL3)

* **Toi41**

(P\_TL4 == green) & (ili4.HaveCarForMe)

Oli1.PopElementWithoutTarget(ili4)

P\_TL4.Move(P\_TL4)

* **Tom41**

(P\_TL4 == green) & (ilm4.HaveCarForMe)

Olm1.PopElementWithoutTarget(ilm4)

P\_TL4.Move(P\_TL4)

* **Toe41**

(P\_TL4 == green) & (ile4.HaveCarForMe)

Ole1.PopElementWithoutTarget(ile4)

P\_TL4.Move(P\_TL4)

* **T\_oli3**

(oli3 != null)

E\_oli3.SendOverNetwork(oli3)

1. **Middle Street:**

**Place Types:**

* **DataCar:** P\_oli2, P1, E\_oli3, P4
* **DataCarQueue:** P1, P3
* **DataTransfer:** I\_i4, I\_i2

**Guards & maps:**

* **T1**

(P\_oli2 != null) & (P1.CanAddCars)

P1.AddElement(P\_oli2)

* **T2**

(P1.HaveCar)

P2.PopElementWithoutTarget(P1)

* **T3**

(P2 != null)

I\_i4.SendOverNetwork(P2)

* **T4**

(E\_oli3 != null) & (P3.CanAddCars)

P3.AddElement(E\_oli3)

* **T5**

(P3.HaveCar)

P4.PopElementWithoutTarget(P3)

* **T6**

(P4 != null)

I\_i2.SendOverNetwork(P4)

1. **Controller:**

**Place Types:**

* **DataString:** ini, in1, in2, red, green, yellow, r1r2r3r4, g1r2r3r4, y1r2r3r4, r1g2r3r4, r1y2r3r4, r1r2g3r4, r1r2y3r4, r1r2r3g4, r1r2r3y4
* **DataTransfer:** op1, op2, op3, op4
* **DataInteger:** Two

**Guards & maps:**

* **Init:**

(init != null)

Ini.SendOverNetwork(OP1)

Ini.SendOverNetwork(OP2)

Ini.SendOverNetwork(OP3)

Ini.SendOverNetwork(OP4)

Ini.MakeNull(ini)

* **T1**

(r1r2r3r4 != null)

g1r2r3r4.Move(r1r2r3r4)

OP1.SendOverNetwork(green)

OP3. SendOverNetwork(green)

* **T2**

(g1r2r3r4 != null) & (in1 == null)

r1r2r3r4.Move(g1r2r3r4)

OP1.SendOverNetwork(yellow)

OP3. SendOverNetwork(yellow)

T2 = DynamicDelay(Two)

* **T3**

(y1r2r3r4 != null)

r1g2r3r4.Move(y1r2r3r4)

OP1.SendOverNetwork(red)

OP3. SendOverNetwork(red)

OP2. SendOverNetwork(green)

OP4. SendOverNetwork(green)

* **T4**

(r1g2r3r4 != null) & (in2 == null)

r1y2r3r4.Move(r1g2r3r4)

OP2. SendOverNetwork(yellow)

OP4. SendOverNetwork(yellow)

T4 = DynamicDelay(Two)

* **T5**

(r1y2r3r4 != null)

r1r2g3r4.Move(r1y2r3r4)

OP2. SendOverNetwork(red)

OP4. SendOverNetwork(red)

OP1. SendOverNetwork(green)

OP3. SendOverNetwork(green)

* **T6**

(r1r2g3r4 != null) & (in1 == null)

r1r2y3r4.Move(r1r2g3r4)

OP1. SendOverNetwork(yellow)

OP3. SendOverNetwork(yellow)

T6 = DynamicDelay(Two)

* **T7**

(r1r2y3r4 != null)

r1r2r3g4.Move(r1r2y3r4)

OP1. SendOverNetwork(red)

OP3. SendOverNetwork(red)

OP2. SendOverNetwork(green)

OP4. SendOverNetwork(green)

* **T8**

(r1r2r3g4 != null) & (in2 == null)

r1r2r3y4.Move(r1r2r3g4)

OP2. SendOverNetwork(yellow)

OP4. SendOverNetwork(yellow)

T8 = DynamicDelay(Two)

* **T9**

(r1r2r3y4 != null)

r1r2r3r4.Move(r1r2r3y4)

OP2. SendOverNetwork(red)

OP4. SendOverNetwork(red)

1. **Component Diagram**
2. **Testing**

* **First Test:**

The first intersection (the Round About), got an input in “i1” place, that exit the round about in “oli2”. From here, it goes through the middle street and enters as an input in the second intersection through the “P\_ai4” input.

A screenshot of a computer

Description automatically generated

PlaceList [i1(Null) e1(Null) i2(Null) e2(Null) i3(Null) e3(Null) oli1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) ole3(Null) ili1|(NULL)| ile1|()| ili2|()| ile2|()| ili3|()| ile3|()| P\_oli2(localhost-1084-P\_oli2)]

ExecutionList [T\_oli2 Temp Marking [oli2(Peugeot-B999BUC)]]

PlaceList [i1(Null) e1(Null) i2(Null) e2(Null) i3(Null) e3(Null) oli1(Null) ole1(Null) **oli2(Peugeot-B999BUC)** ole2(Null) oli3(Null) ole3(Null) ili1|(NULL)| ile1|()| ili2|()| ile2|()| ili3|()| ile3|()| P\_oli2(localhost-1084-P\_oli2)]

ExecutionList [Ti17 Temp Marking []]

PlaceList [i1(Null) e1(Null) i2(Null) e2(Null) i3(Null) e3(Null) oli1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) ole3(Null) ili1|(ili1(Peugeot-B999BUC))| ile1|()| ili2|()| ile2|()| ili3|()| ile3|()| P\_oli2(localhost-1084-P\_oli2)]

ExecutionList [Ti17 Temp Marking []]

PlaceList [i1(Null) e1(Null) i2(Null) e2(Null) i3(Null) e3(Null) oli1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) ole3(Null) **ili1|(ili1(Peugeot-B999BUC))|** ile1|()| ili2|()| ile2|()| ili3|()| ile3|()| P\_oli2(localhost-1084-P\_oli2)]

ExecutionList [Ti11 Temp Marking [i1(Peugeot-B999BUC)]]

PlaceList [**i1(Peugeot-B999BUC)** e1(Null) i2(Null) e2(Null) i3(Null) e3(Null) oli1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) ole3(Null) ili1|()| ile1|()| ili2|()| ile2|()| ili3|()| ile3|()| P\_oli2(localhost-1084-P\_oli2)]

$$$$$$$$$$$$$$$ I got an Input From NetWork for i1

PlaceList [P\_oli2(Null) P1|(NULL)| P2(Null) I\_i4(localhost-1082-P\_ai4) E\_oli3(Null) P3|()| P4(Null) I\_i2(localhost-1081-i2)]

ExecutionList [T3 Temp Marking [P2(Peugeot-B999BUC)]]

PlaceList [P\_oli2(Null) P1|(NULL**)| P2(Peugeot-B999BUC)** I\_i4(localhost-1082-P\_ai4) E\_oli3(Null) P3|()| P4(Null) I\_i2(localhost-1081-i2)]

ExecutionList [T2 Temp Marking []]

PlaceList [P\_oli2(Null) P1|(P1(Peugeot-B999BUC))| P2(Null) I\_i4(localhost-1082-P\_ai4) E\_oli3(Null) P3|()| P4(Null) I\_i2(localhost-1081-i2)]

ExecutionList [T1 Temp Marking [P\_oli2(Peugeot-B999BUC)]]

PlaceList [**P\_oli2(Peugeot-B999BUC)** P1|()| P2(Null) I\_i4(localhost-1082-P\_ai4) E\_oli3(Null) P3|()| P4(Null) I\_i2(localhost-1081-i2)]

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_oli2

ConstantPlaceList [full(full)]

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) P\_am1(Null) ilm1|()| P\_bm1(Null) P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(Null) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2|()| P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(Null) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(Null) OP3(localhost-1082-in1) P\_ai4(Null) **ili4|(ili4(Peugeot-B999BUC))|** P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(Null) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) olm3(Null) ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

ExecutionList [T\_ai4 Temp Marking [P\_ai4(Peugeot-B999BUC)]]

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) P\_am1(Null) ilm1|()| P\_bm1(Null) P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(Null) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2|()| P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(Null) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(Null) OP3(localhost-1082-in1**) P\_ai4(Peugeot-B999BUC)** ili4|()| P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(Null) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) olm3(Null) ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_ai4

* **Second Test:**

The second test was performed to test how the second intersection works with the controller. It receives an input from “P\_am1”, enters the queue in “ilm1” and waits for the green light at the traffic light. Then, it enters the “ilm2” queue and waits for the second traffic light to turn green. After, it exists the intersection through the “olm3” place.

A screenshot of a computer

Description automatically generated

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) P\_am1(Null) ilm1|(NULL)| P\_bm1(Null) P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(red) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2|(NULL)| P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(green) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(red) OP3(localhost-1082-in1) P\_ai4(Null) ili4|()| P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(green) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) **olm3(Mazda-B999BUC)** ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

ExecutionList [T\_bm2 Temp Marking [P\_TL2(green)]]

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_TL4

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_TL2

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_TL3

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_TL1

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) P\_am1(Null) ilm1|(NULL)| P\_bm1(Null) P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(green) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2**|(ilm2(Mazda-B999BUC))|** P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(red) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(green) OP3(localhost-1082-in1) P\_ai4(Null) ili4|()| P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(red) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) olm3(Null) ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

ExecutionList [T\_am12 Temp Marking [P\_bm1(Mazda-B999BUC)]]

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) P\_am1(Null) ilm1|(NULL)| **P\_bm1(Mazda-B999BUC)** P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(green) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2|()| P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(red) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(green) OP3(localhost-1082-in1) P\_ai4(Null) ili4|()| P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(red) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) olm3(Null) ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) P\_am1(Null) **ilm1|(ilm1(Mazda-B999BUC)**)| P\_bm1(Null) P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(Null) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2|()| P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(red) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(green) OP3(localhost-1082-in1) P\_ai4(Null) ili4|()| P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(red) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) olm3(Null) ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

ExecutionList [T\_bm1 Temp Marking [P\_TL1(green)]]

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) P\_am1(Null) **ilm1|(ilm1(Mazda-B999BUC))|** P\_bm1(Null) P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(red) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2|()| P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(red) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(red) OP3(localhost-1082-in1) P\_ai4(Null) ili4|()| P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(red) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) olm3(Null) ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

ExecutionList [T\_am1 Temp Marking [P\_am1(Mazda-B999BUC)]]

PlaceList [P\_ai1(Null) ili1|()| P\_bi1(Null) **P\_am1(Mazda-B999BUC)** ilm1|()| P\_bm1(Null) P\_ae1(Null) ile1|()| P\_be1(Null) OP1(localhost-1082-in1) P\_TL1(red) P\_ai2(Null) ili2|()| P\_am2(Null) ilm2|()| P\_ae2(Null) ile2|()| OP2(localhost-1082-in2) P\_TL2(red) P\_ai3(Null) ili3|()| P\_am3(Null) ilm3|()| P\_ae3(Null) ile3|()| P\_TL3(red) OP3(localhost-1082-in1) P\_ai4(Null) ili4|()| P\_am4(Null) ilm4|()| P\_ae4(Null) ile4|()| OP4(localhost-1082-in2) P\_TL4(red) oli1(Null) olm1(Null) ole1(Null) oli2(Null) ole2(Null) oli3(Null) olm3(Null) ole3(Null) oli4(Null) olm4(Null) ole4(Null) P\_ci1(Null) P\_di1(Null) P\_dm1(Null) P\_de1(Null) P\_ei1(Null) P\_em1(Null) P\_ee1(Null) E\_oli3(localhost-1084-E\_oli3)]

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_TL4

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_am1

$$$$$$$$$$$$$$$ I got an Input From NetWork for P\_TL3