PYTHON ROBOTICS

WORKSHOP 4

HUISWERK

- Maak wandvolgen werkend.
- Stop als minder dan 300mm van de achterwand bent.
- Geef de status van de regeling weer via de led's.

Bonuspunten:

- Maak wandvolgen beter.

Tip: gebruik een lange USB-kabel om met REPL te volgen wat geprint wordt.

HUISWERK

- Maak wandvolgen werkend.
- Stop als minder dan 300mm van de achterwand bent.
- Geef de status van de regeling weer via de led's.

Bonuspunten:

- Maak wandvolgen beter.

Tip: gebruik een lange USB-kabel om met REPL te volgen wat geprint wordt.

GLOBAL VARIABELEN

```
1  def Test():
2      print("Test1", aap)
3
4  aap = 7
5  Test()
6
7
8
```

GLOBAL VARIABELEN

```
1  def Test():
2      print("Test1", aap)
3
4  aap = 7
5  Test()
6
7
8
```

```
9  def Test():
10     aap = 8
11     print("Test1", aap)
12
13 aap = 7
14 Test()
15 print("Test2", aap)
16
```

GLOBAL VARIABELEN

```
1  def Test():
2      print("Test1", aap)
3
4  aap = 7
5  Test()
6
7
8
```

```
9  def Test():
10     aap = 8
11     print("Test1", aap)
12
13 aap = 7
14 Test()
15 print("Test2", aap)
16
```

```
17 # error
18 def Test():
19     aap = aap + 1
20     print("Test1", aap)
21
22 aap = 7
23 Test()
24
```

```
>>> Traceback (most recent call last):
  File "main.py", line 23, in <module>
  File "main.py", line 19, in Test
NameError: local variable referenced before assignment
```

GLOBAL VARIABELEN

```
1  def Test():
2      print("Test1", aap)
3
4  aap = 7
5  Test()
6
7
8
```

```
9  def Test():
10     aap = 8
11     print("Test1", aap)
12
13 aap = 7
14 Test()
15 print("Test2", aap)
16
```

```
17 # error
18 def Test():
19     aap = aap + 1
20     print("Test1", aap)
21
22 aap = 7
23 Test()
24
```

```
25 def Test():
26     global aap
27     aap = aap + 1
28     print("Test1", aap)
29
30 aap = 7
31 Test()
32 print("Test2", aap)
```

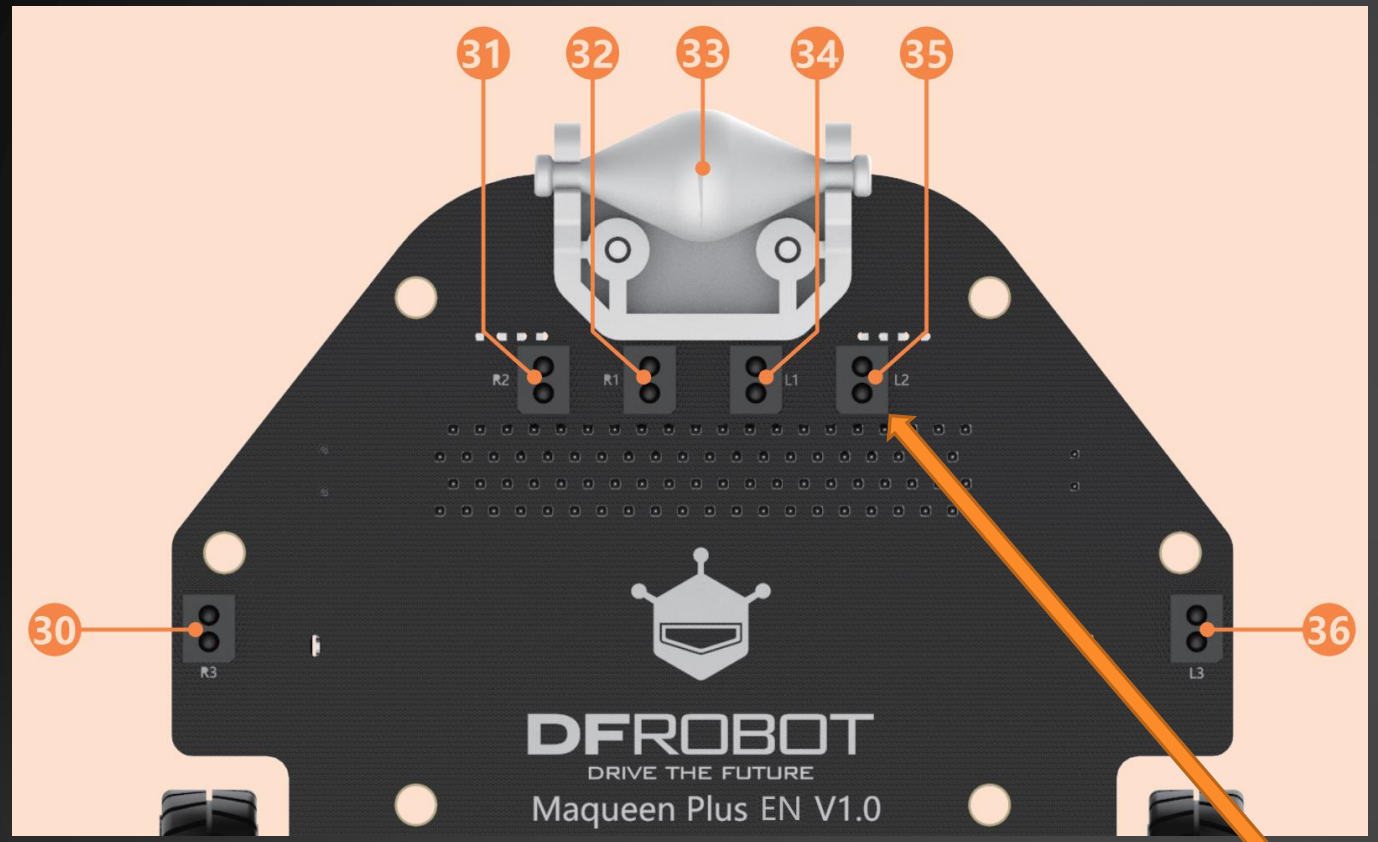
VARIABELE SCOPE

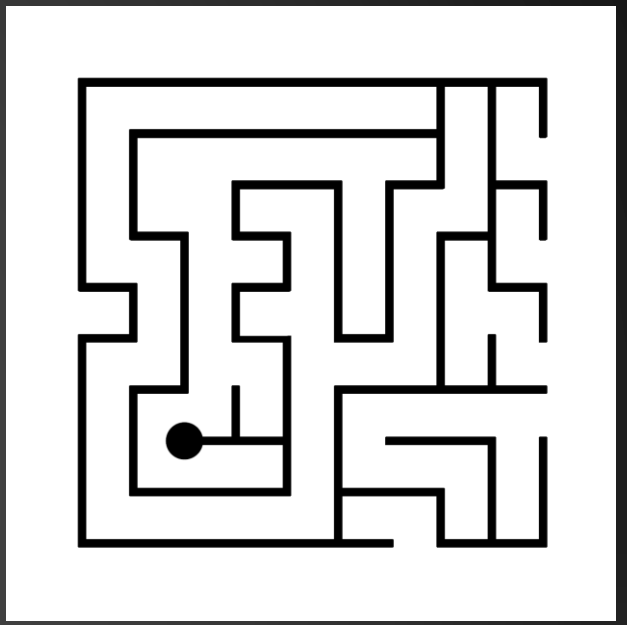
- Je kunt in functies globale variabelen lezen.
- Variabelen waar je in functies iets aan toekent, zijn default lokaal.
- Met het keyword ‘global’ in een functie geef je aan dat een variabele niet lokaal is.

MYQUEEN API

• GetEncoders()	return: list met waarde van linkse en rechtse encoder
• GetFloorSensors()	return: list met 6 analoge waarden, link->rechts
• RGB(ColourL, ColourR)	Stuur RGB Leds, kleurenbits: 1=rood, 2=groen, 4=blauw
• Servo(Nr, Degrees)	Stuur servo, Nr = 1..3
• Motors(SpeedL, SpeedR)	Cmd: stuur motors met gegeven snelheid
• Rotate(Degrees, Speed)	Cmd: roteer robot 'Degrees' graden met gevraagde snelheid
• SpeedDistance(Speed, Distance)	Cmd: Rij gegeven afstand met gevraagde snelheid
• Stop()	Cmd: Stop robot
• IsDone()	Takt & controle status, geeft True terug als commando klaar

VLOERSENSOR

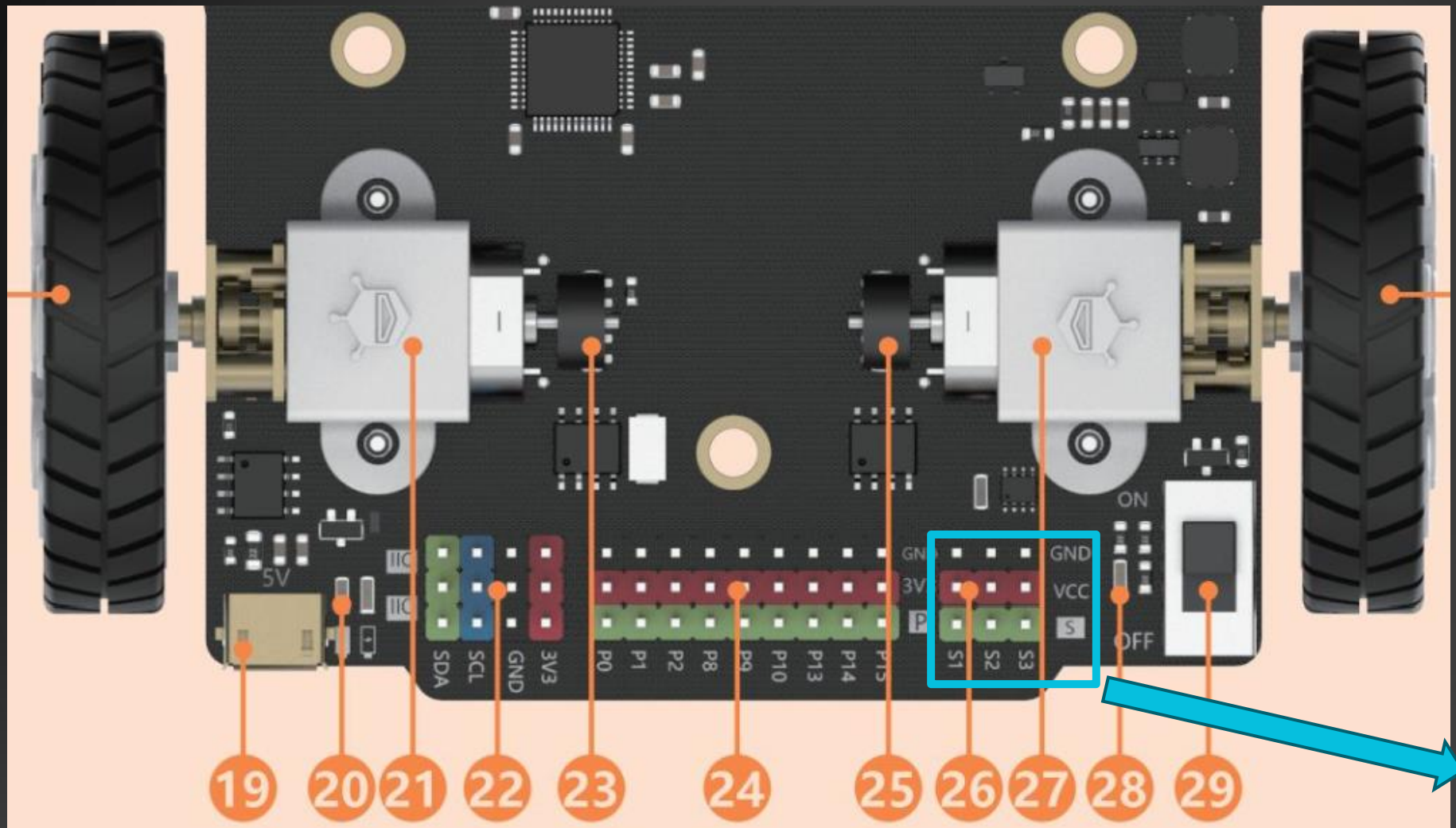




Op lichte ondergrond:

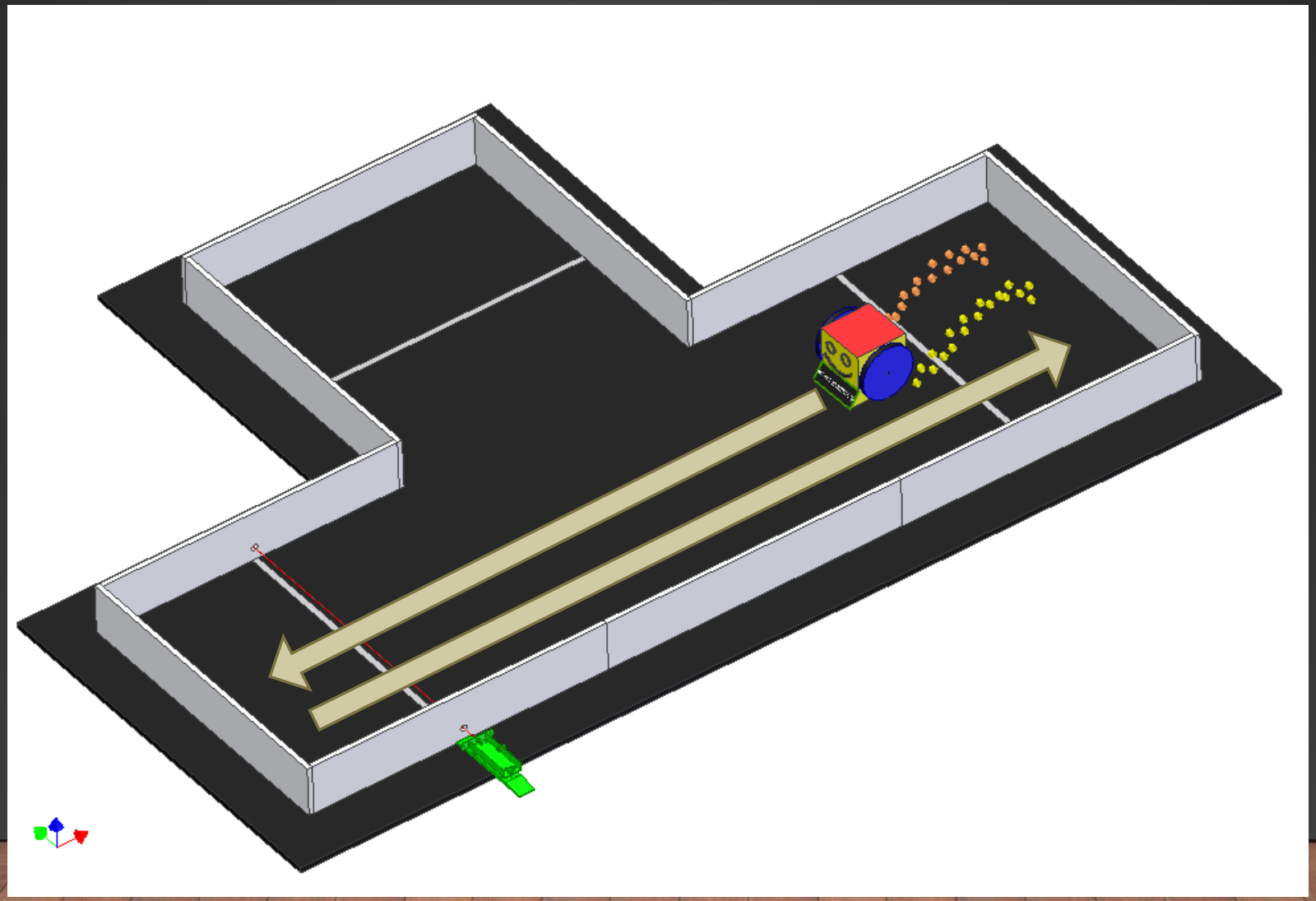
```
>>>
>>> f = Mq.GetFloorSensors()
>>> f
(3677, 3330, 3786, 3798, 3769, 3634)
>>> print(f[1])
3330
>>>
```

SERVO'S




```
>>> Mq.Servo(3, 0)
>>> Mq.Servo(3, 170)
```

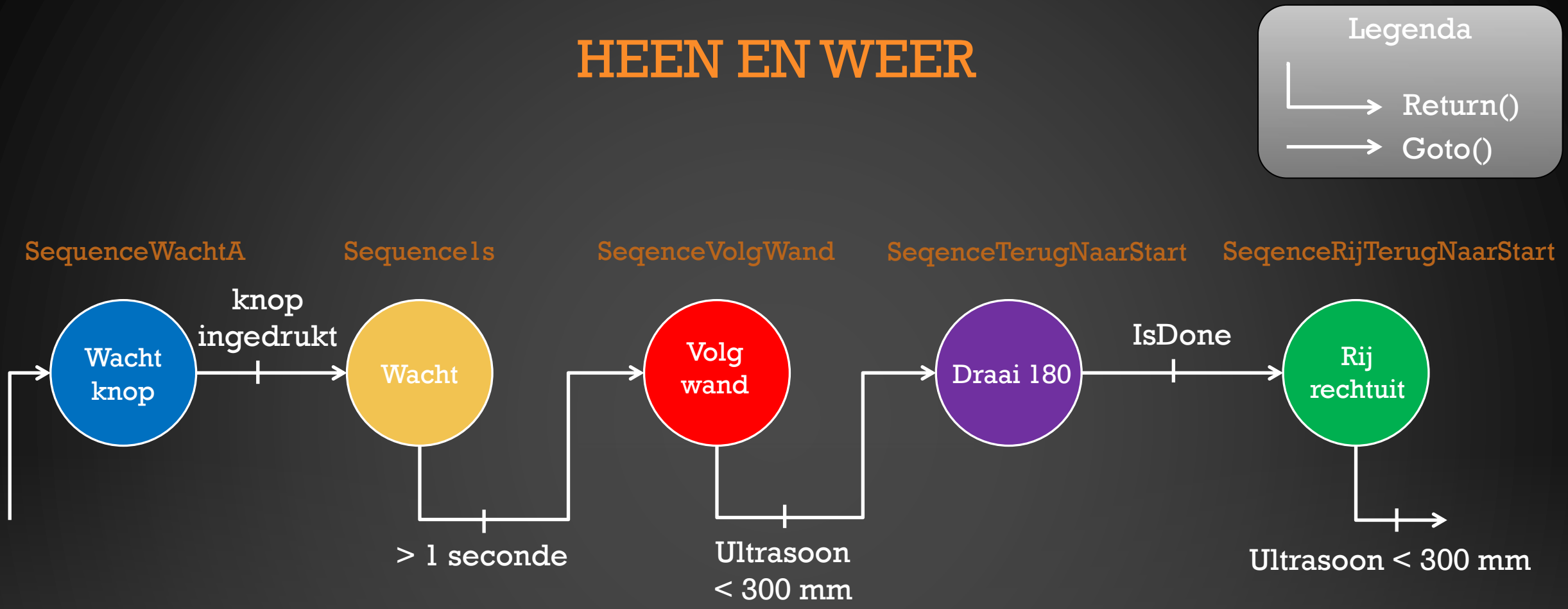
HEEN EN WEER – DE OPGAVE



HEEN EN WEER – TWEE NIEUWE STATES

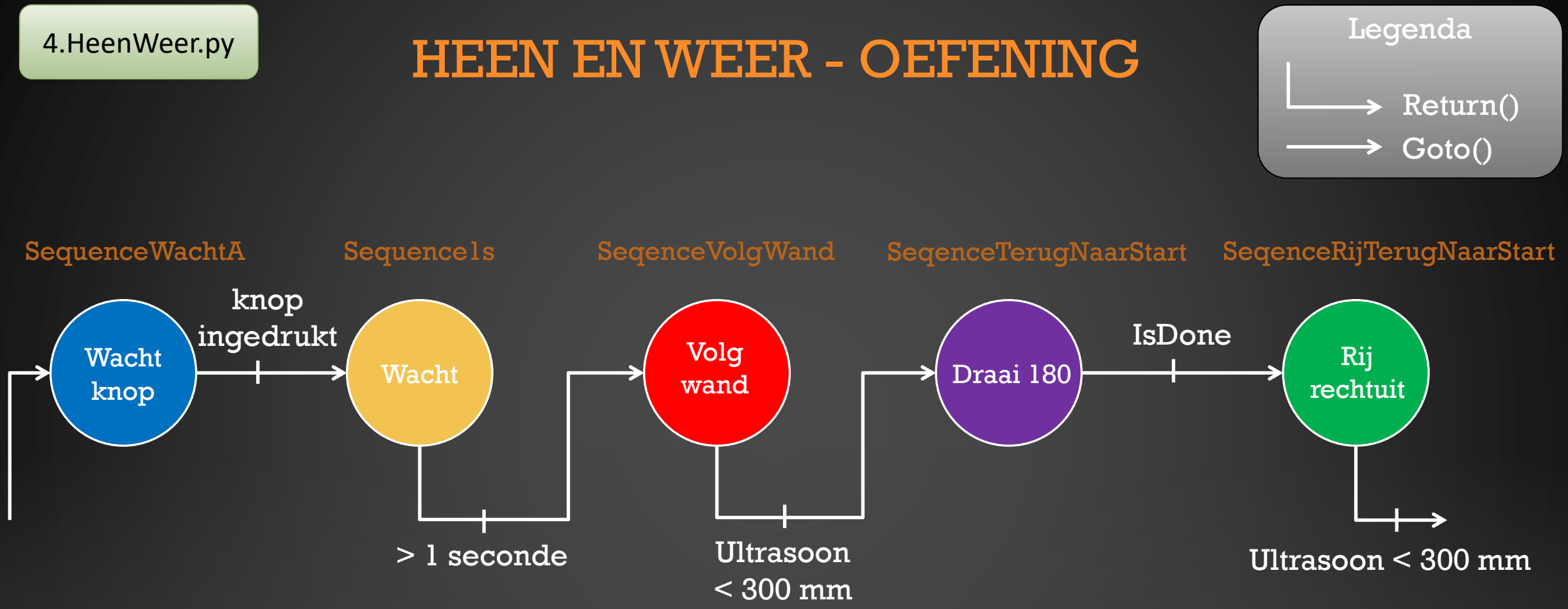
```
39 # ---------------------------------------------------------------
40 def SeqenceTerugNaarStart(S):
41     if S.IsNewState('SeqenceReturnToStart') :
42         Mq.RGB(3, 3)
43         Mq.Rotate(180, 40)    # draai 180 graden
44 
45     if Mq.IsDone() :
46         S.Goto(SeqenceRijTerugNaarStart)
47 
48 # ---------------------------------------------------------------
49 def SeqenceRijTerugNaarStart(S):
50     if S.IsNewState('SeqenceRijTerugNaarStart') :
51         Mq.RGB(2, 2)
52         Mq.SpeedDistance(100, 9999) # rechtuit rijden, heel ver
53 
54     # stop als we dichtbij een obstakel (wand A) zien
55     if Sensoren.UsAfstand < 300 :
56         Mq.Stop()
57         S.Return()
58 
59     Mq.IsDone() # Dit zorgt er voor dat SpeedDistance goed werkt
```

HEEN EN WEER


Legenda
Return()
Goto()
SequenceWachtA
Sequence1s
SeqenceVolgWand
SeqenceTerugNaarStart
SeqenceRijTerugNaarStart
Wacht knop
knop ingedrukt
Wacht
> 1 seconde
Volg wand
Ultrasoon < 300 mm
Draai 180
IsDone
Rij rechtuit
Ultrasoon < 300 mm

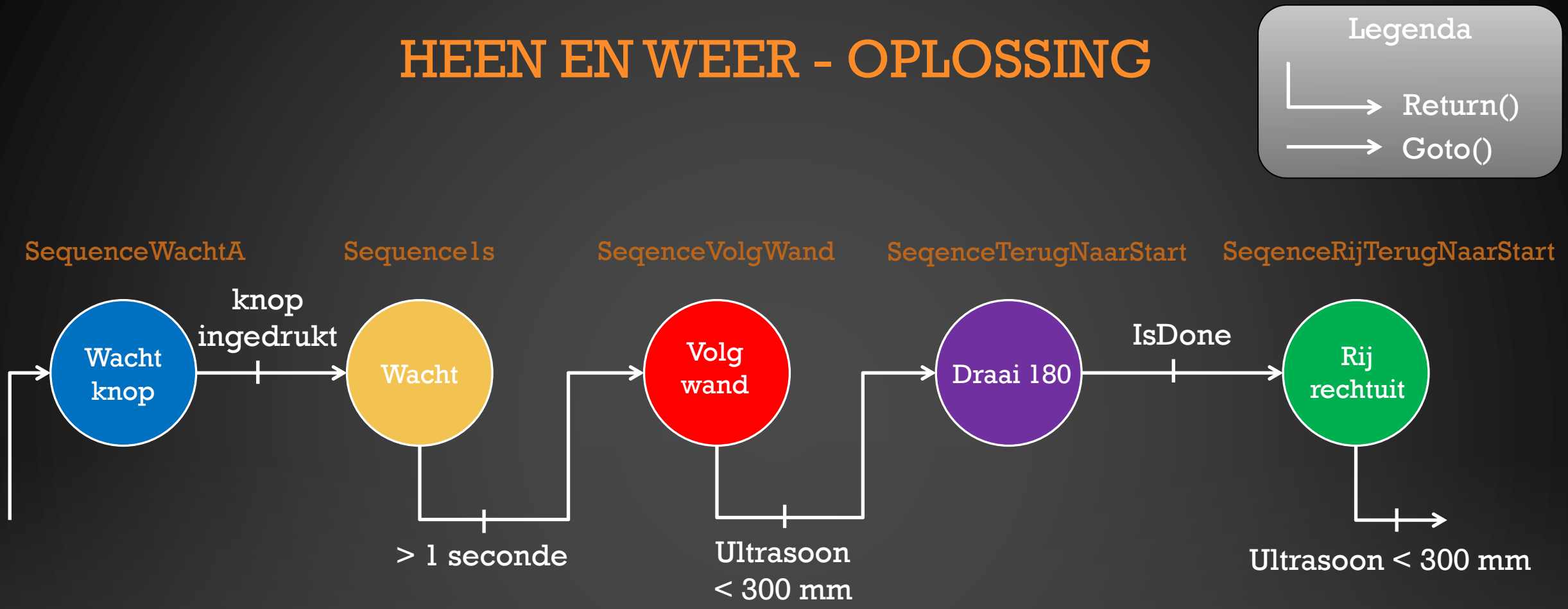
4.HeenWeer.py

HEEN EN WEER - OEFENING




```
69
70  Sm.Goto( vul_dit_in )
71
```

HEEN EN WEER - OPLOSSING




```
69
70 Sm.Goto([SequenceWachtA, SeqenceVolgWand, SeqenceTerugNaarStart])
71
```

WHILE LOOP

```
1  i = 1
2  while i < 6:
3    print(i)
4    i += 1
```

```
Running: tmp.py
1
2
3
4
5
>>>
```

BREAK, CONTINUE

```
1  i = 1
2  while i < 6:
3    print(i)
4    if i == 3:
5      break
6    i += 1
```

```
8  i = 0
9  while i < 6:
10   i += 1
11   if i == 3:
12     continue
13   print(i)
```

FOR LOOP

Loop door ‘Containers’

```
1 fruits = ["apple", "banana", "cherry"]
2 for x in fruits:
3   print(x)
```

FOR (BREAK)

```
4 fruits = ["apple", "banana", "cherry"]
5 for x in fruits:
6   print(x)
7   if x == "banana":
8     break
```

RANGE()

```
1  for x in range(6):
2    print(x)
3
4  for x in range(2, 6):
5    print(x)
6
7  for x in range(2, 30, 3):
8    print(x)
```

https://www.w3schools.com/python/

DICTIONARY

```
1  OdoConfig = {
2    "TickToDistance": 0.0353 ,
3    "TickCorrectLR": 1.00073552,
4    "TickToHeading": 0.00015841
5  }
```

Opslaan op je pc:

```
with open('data.json', 'w') as fp:
    json.dump(OdoConfig, fp)
```

```
7  print(OdoConfig)
8
9  print(OdoConfig['TickToDistance'])
10
11 print(OdoConfig.keys())
```

```
{'TickToDistance': 0.0353, 'TickCorrectLR': 1.00073552, 'TickToHeading': 0.00015841}
0.0353
dict_keys(['TickToDistance', 'TickCorrectLR', 'TickToHeading'])
```

LISTIGE DATATYPES OP EEN RIJ

- overzicht van datatypes, aantal zijn we al tegengekomen.

```
>>> MyList = [1, 2, "aap"]
>>>
>>> MyTuple = (9, 10, "Noot")
>>>
>>> MySet = {"Boom", "Roos", "Vis"}
>>>
>>>
>>> MyDict = {"Boom": 11, "Roos": 13, "Vis": 17}
```

	Mutable	Ordered	Indexing / Slicing	Duplicate Elements
List	✓	✓	✓	✓
Tuple	✗	✓	✓	✓
Set	✓	✗	✗	✗

https://towardsdatascience.com/15-examples-to-master-python-lists-vs-sets-vs-tuples-d4ffb291cf07

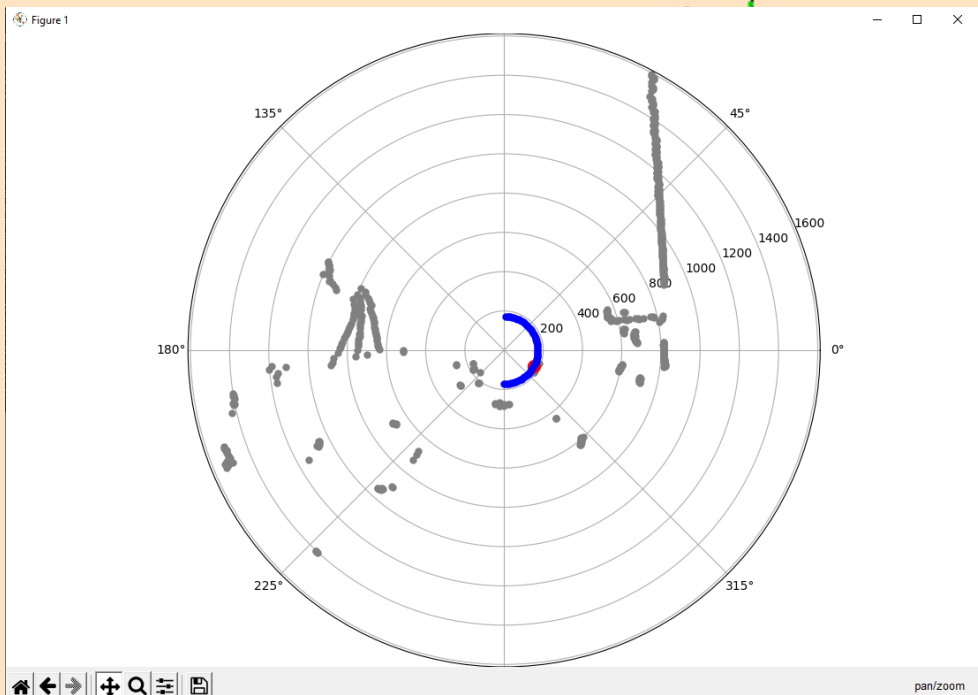
Bridge.bat

```
RlCommsClient UPD, reverse: True
   Publish:   5005
   Subscribe: 5006
Windows
Connect to  \\.\com3 @ 115200 bps
```

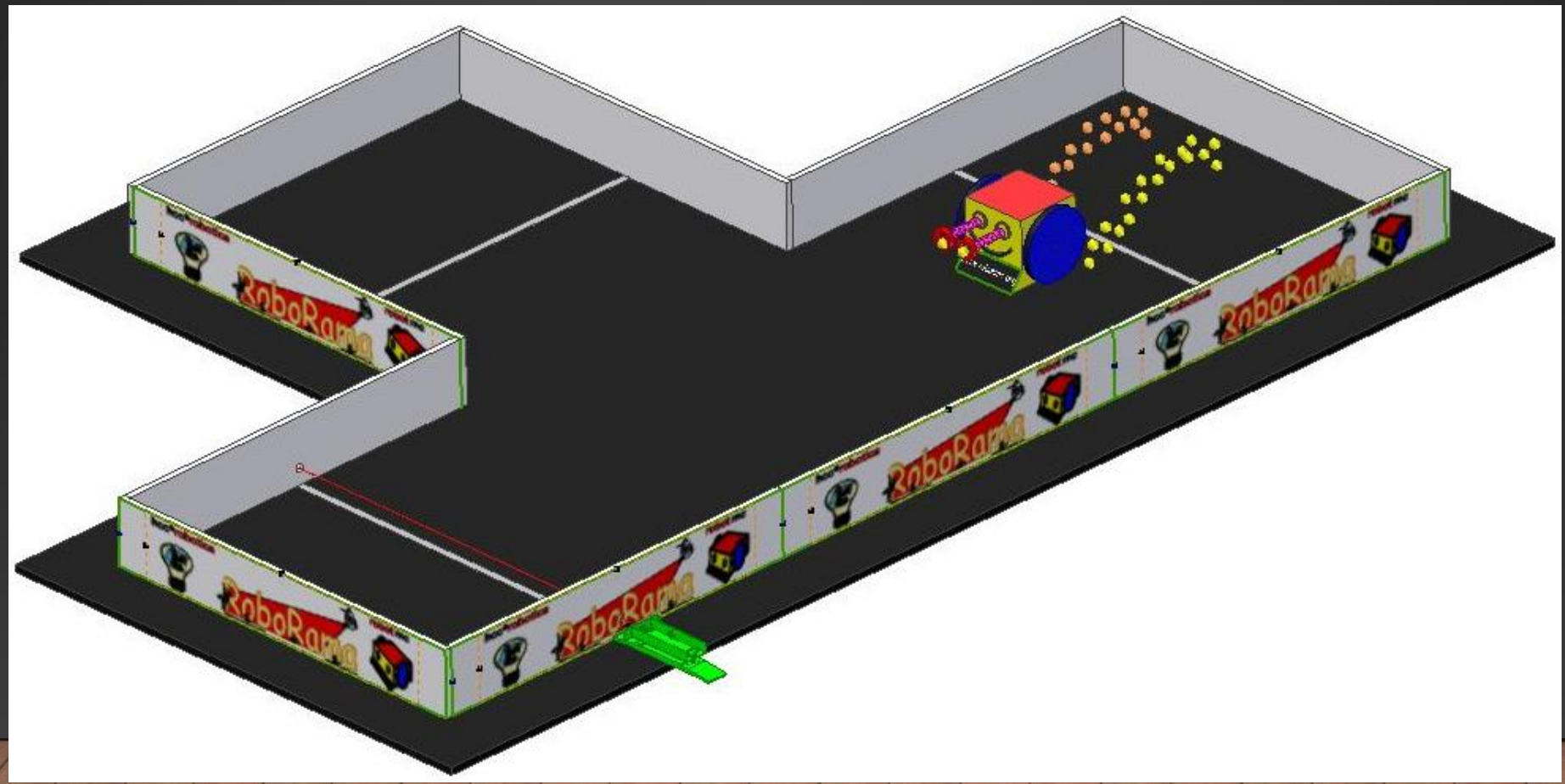
Terminal.py

Quit | LogStart | SaveAll | Upload | Player | Abort | Frames

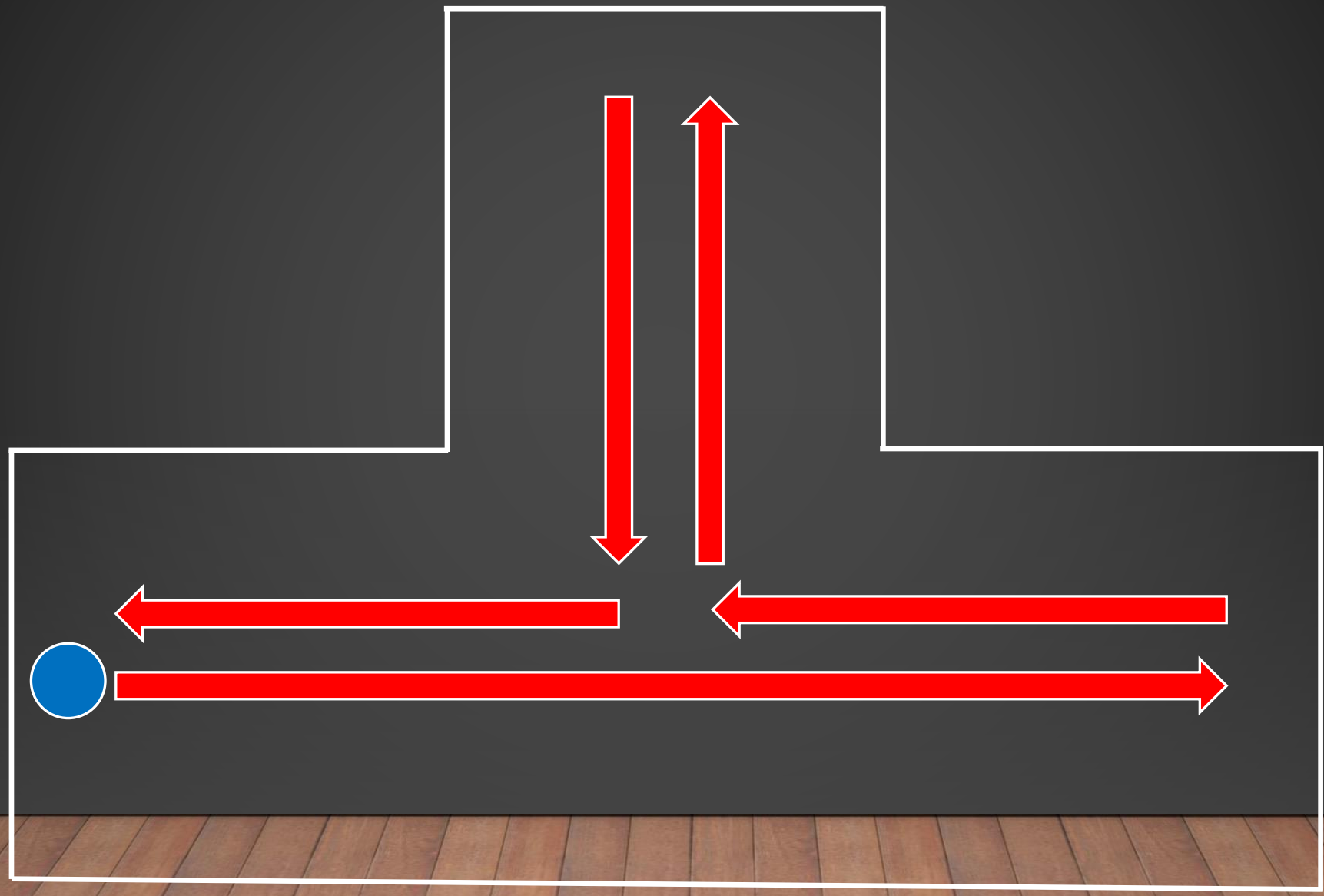
```
[14:I] UmSet XY - X: 3000, Y: 900, Speed: 100, EndSpeed: 150
ÁMARKS 0 1273,894À
ÁMARKS 0 1308,660À
SSFindPassage Heading: -90, Range: -88, Result: -82, ret: 82
[14:W] Start movement XY, params X: 3000, Y: 900, Speed: 100, EndSpeed: 150
ÁPOSITION 1284 893 -0.076549À
ÁMARKS 0 1315,657À
SSFindPassage Heading: -90, Range: -88, Result: -83, ret: 83
ÁPOSITION 1294 893 -0.067708À
ÁMARKS 0 1321,650À
SSFindPassage Heading: -90, Range: -88, Result: -84, ret: 84
ÁPOSITION 1304 892 -0.063300À
```



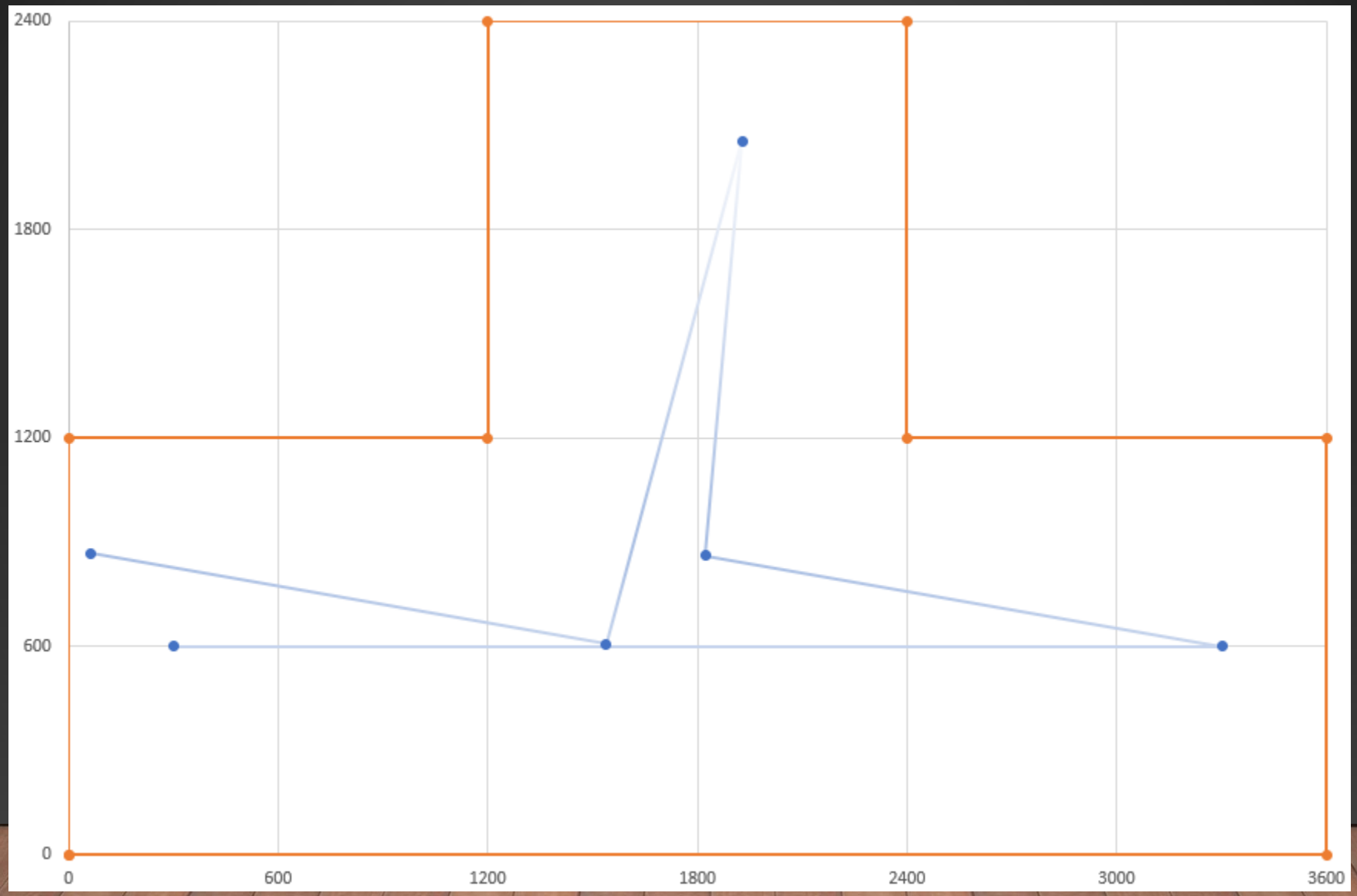
ROBORAMA T-TIJD



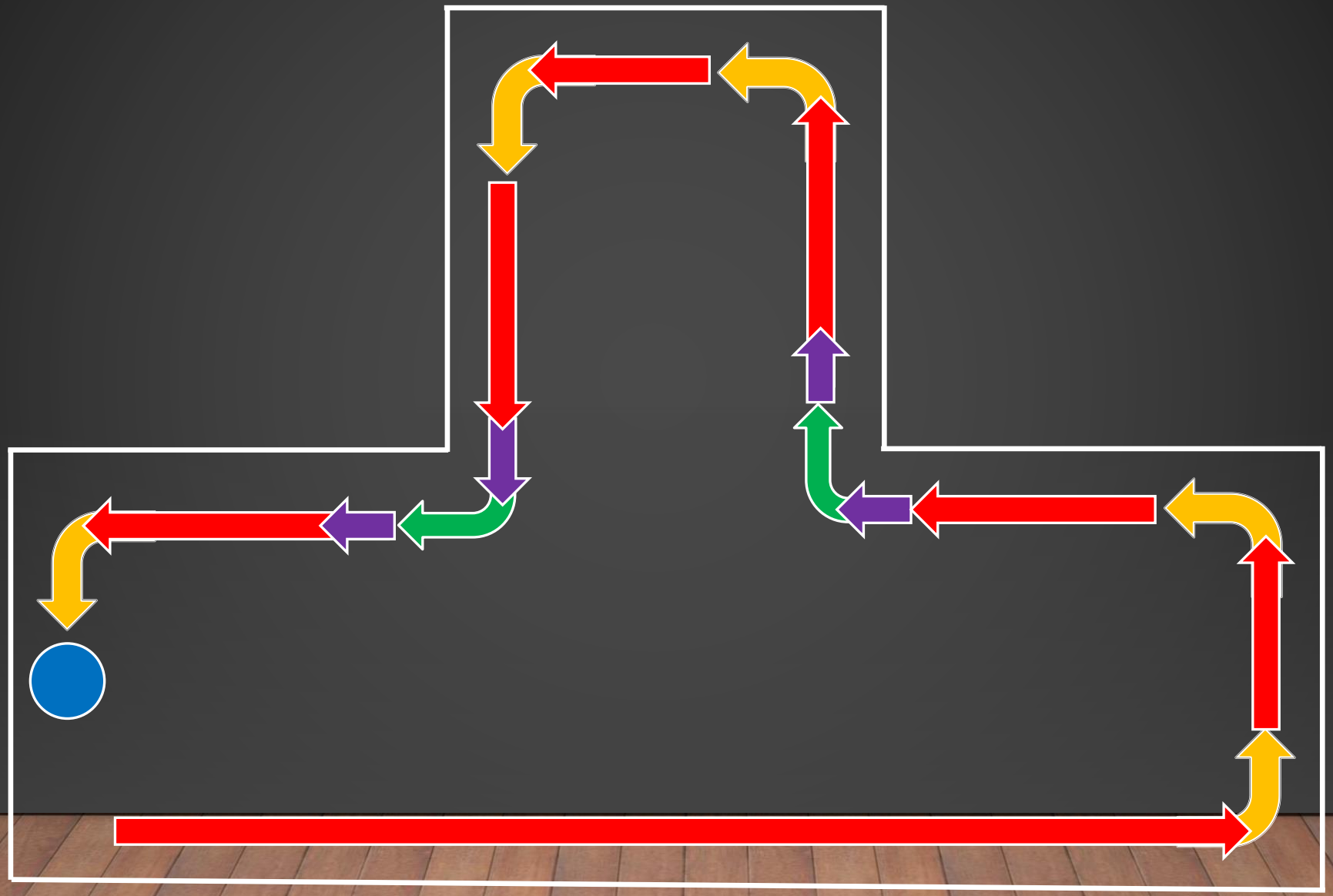
T-TIJD ROUTE



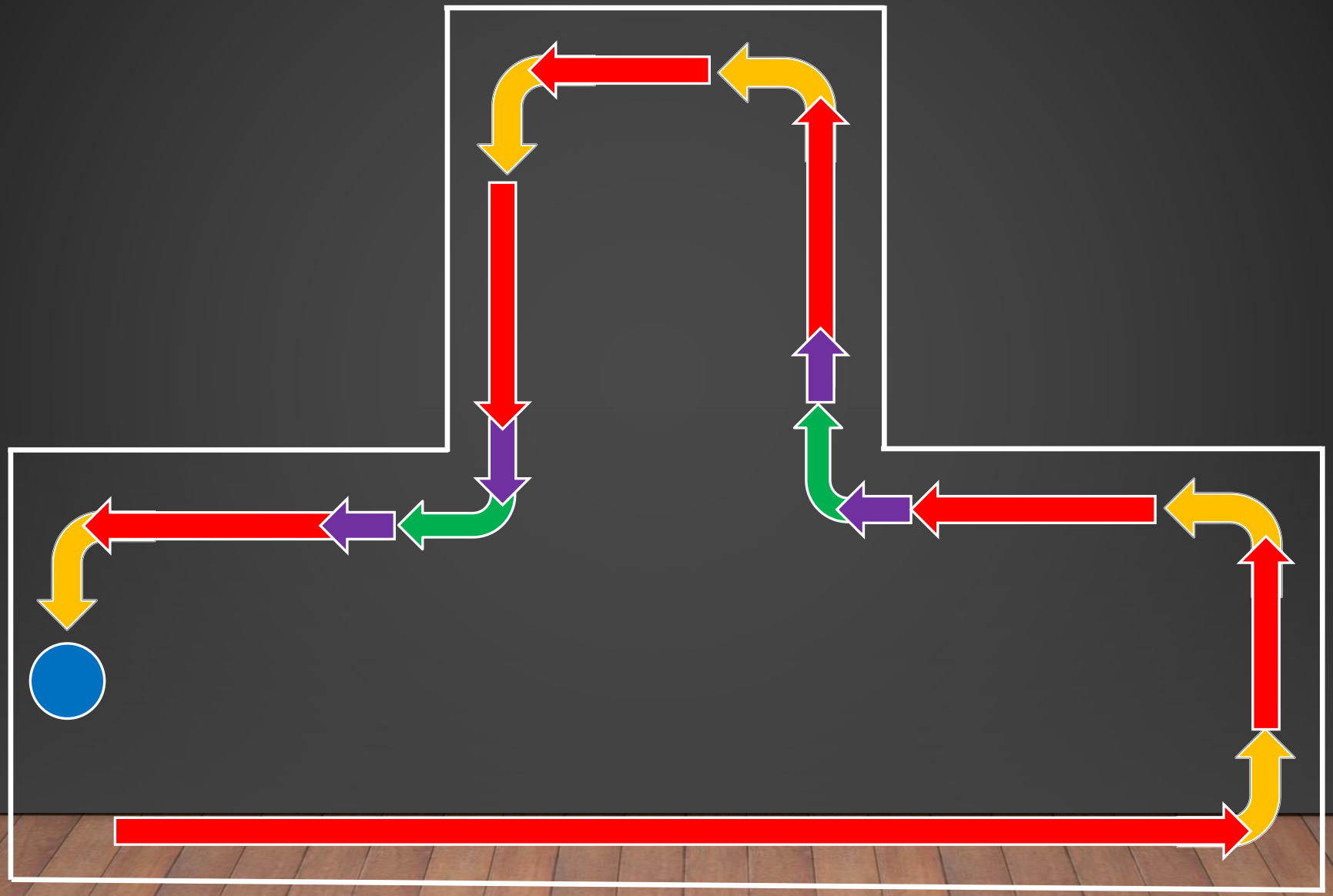
SIMULATIE MET 85 GRADEN I.P.V. 90



T-TIJD STATES



T-TIJD STATES



Volg wand

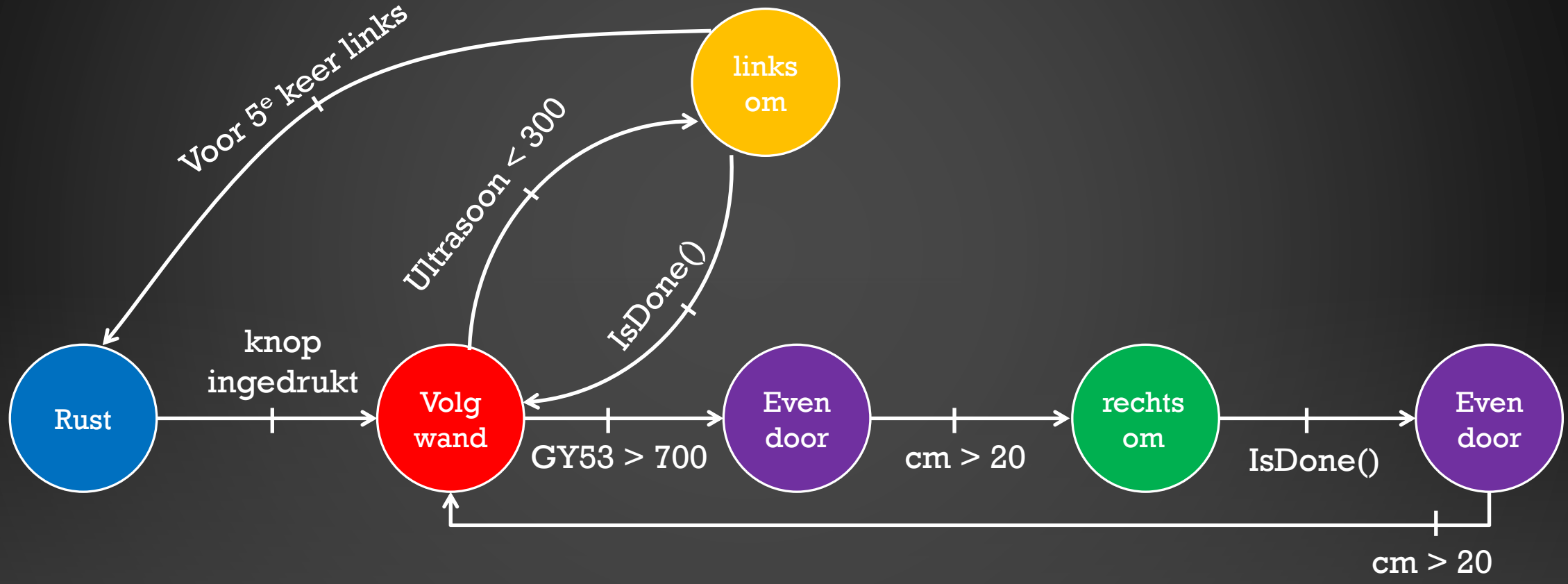
links om

Even door

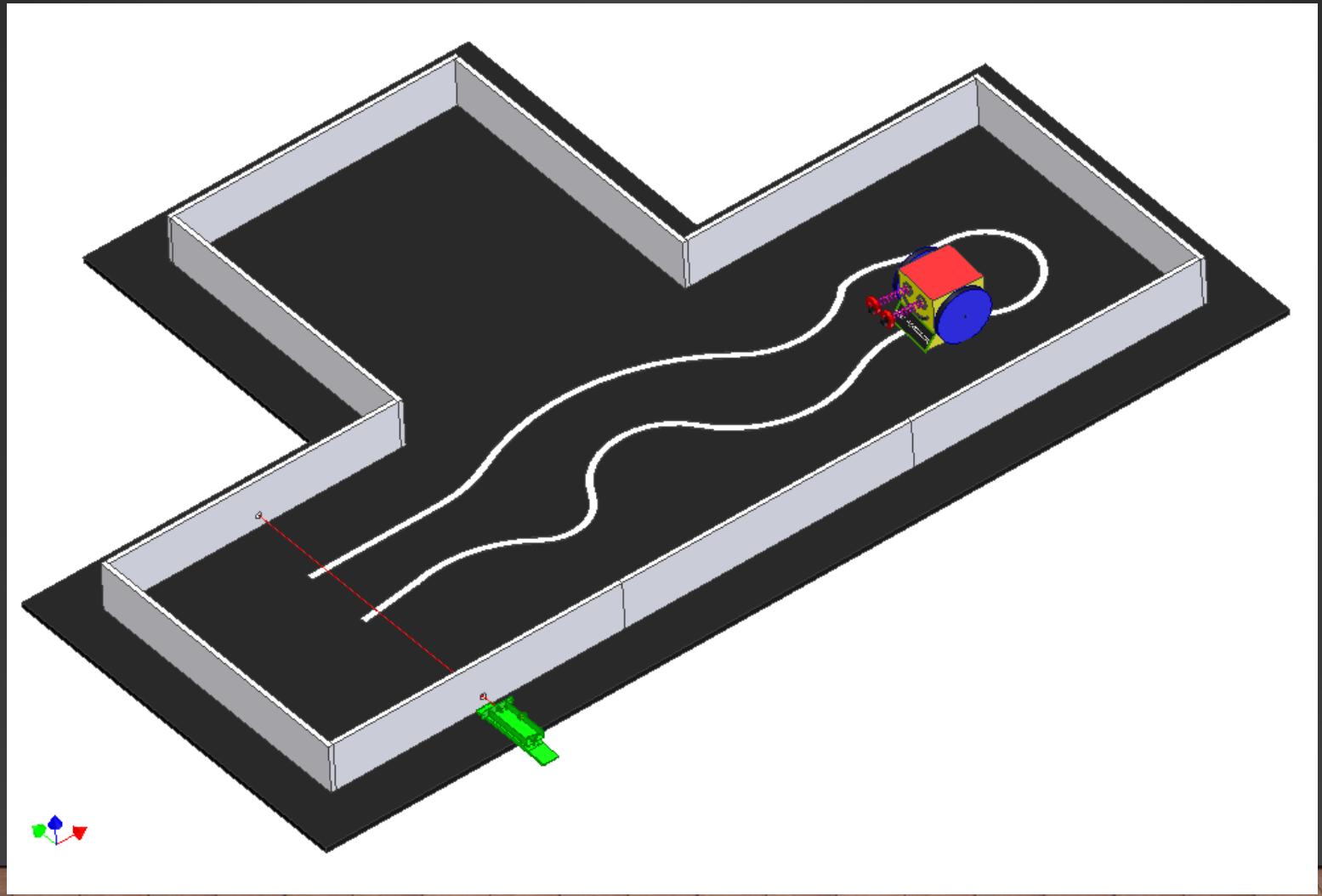
rechts om

Rust

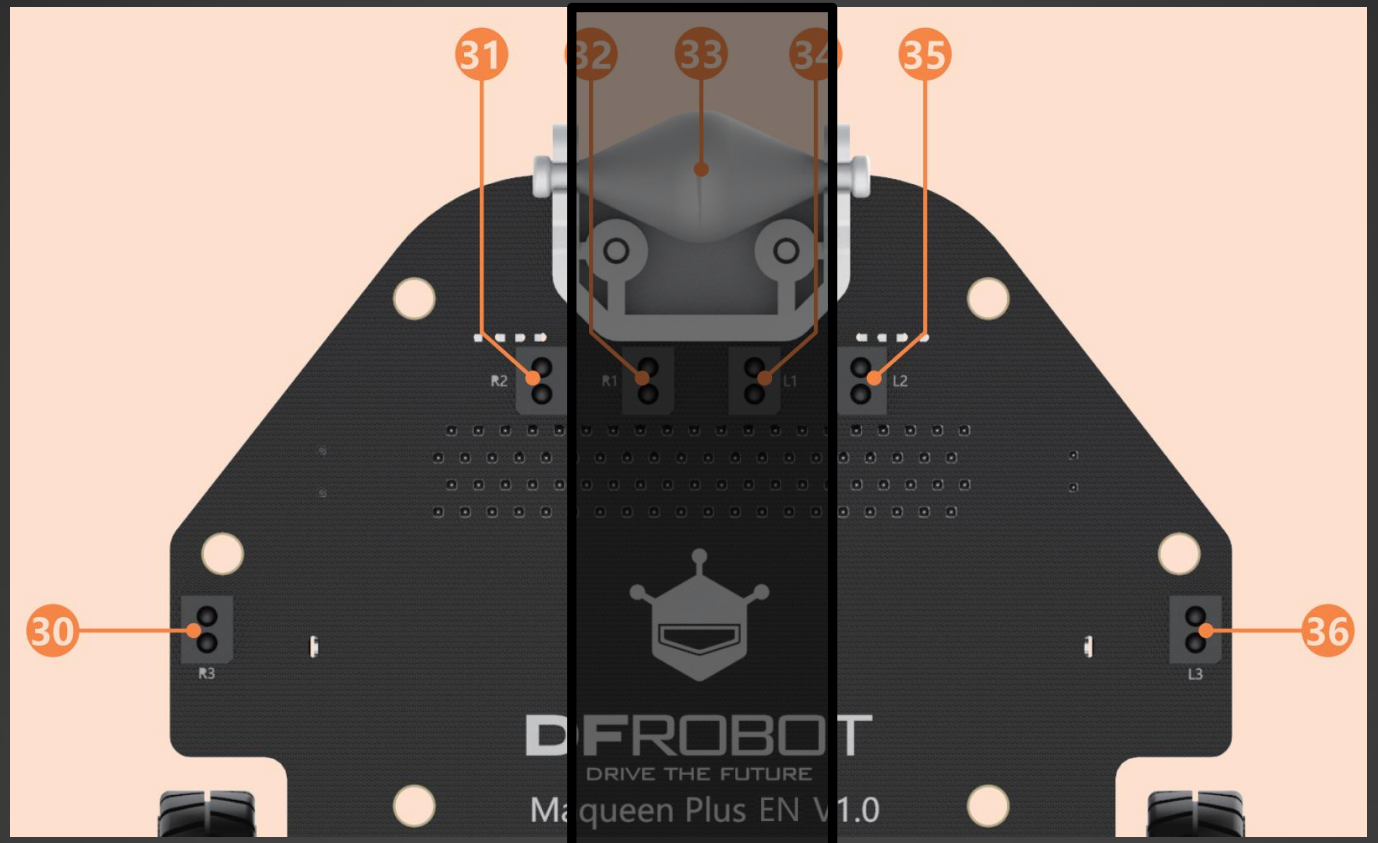
T-TIJD STATE MACHINE



ROBORAMA LIJNVOLGEN

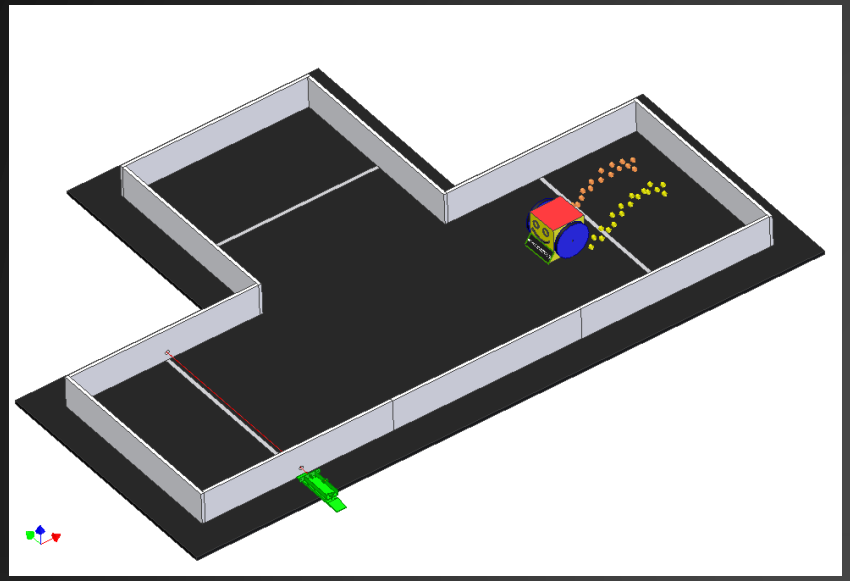


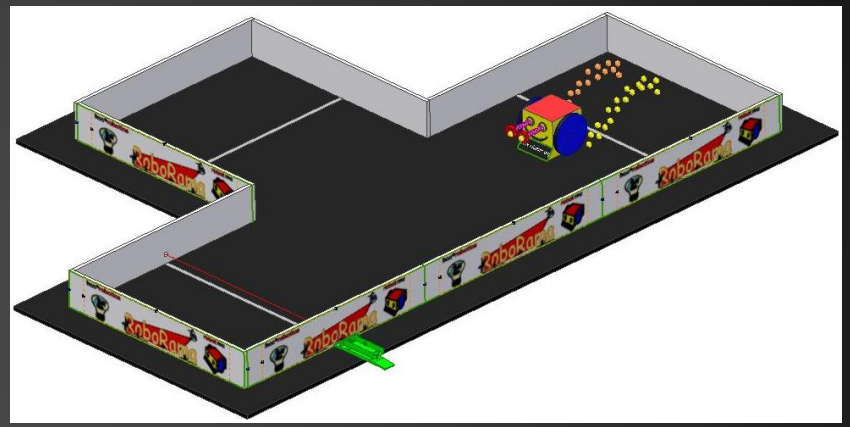
LIJNVOLGEN

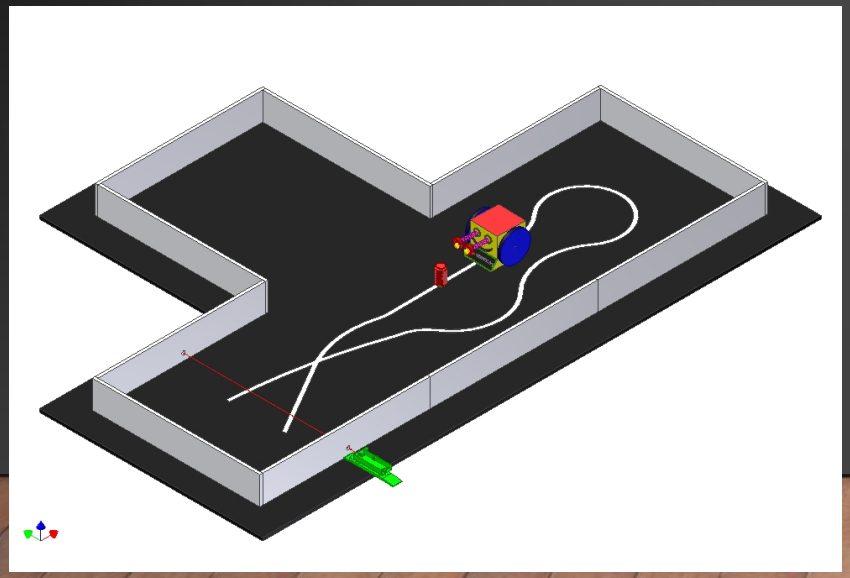

31
32
33
34
35
R2
R1
L1
L2
30
R3
L3
36
DFROBOT
DRIVE THE FUTURE
Maqueen Plus EN V1.0

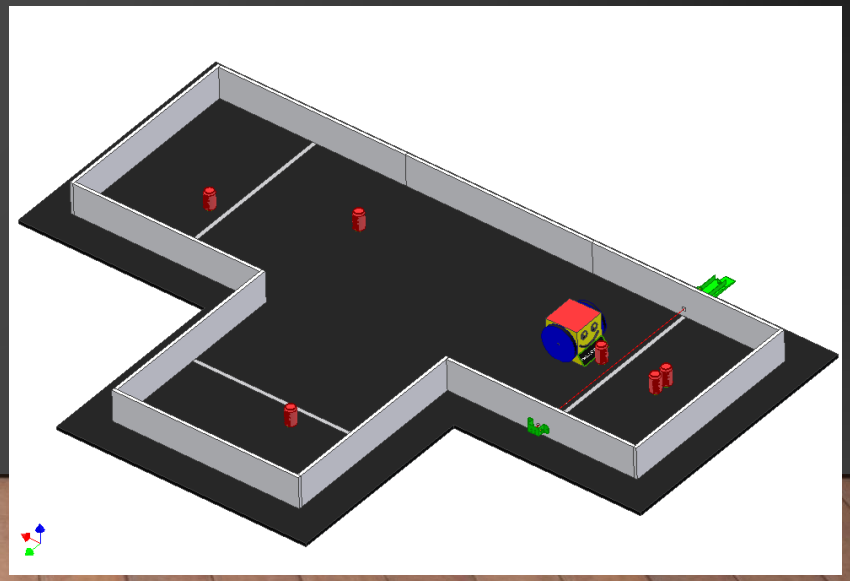
GEBRUIK LIJNSENSOR

ROBORAMA – 22 APRIL 2023









PYTHON ROBOTICS

EINDE…