first\_call = True

02     while driver.step() != -1:

03

04         now = driver.getTime()

05

06         if parking:

07             # jeżeli z klawiatury włączono parkowanie, to:

08

09             if now - last\_sensor\_time >= SENSOR\_INTERVAL:

10                 # co SENSOR\_INTERVAL (zadany na początku) wykonujemy:

11                 last\_sensor\_time = now

12                 if first\_call:

13                     #plotter = palg.LivePlotter(ax\_live)

14                     parker = palg.Parking(driver, "left", now)

15                     yaw\_init = imu.getRollPitchYaw()[2]

16                 first\_call = False

17                 # odczyt odległości

18                 dists = [process\_distance\_sensors(s) for s in distance\_sensors]

19                 names = dict(zip(sensor\_names, dists))

20

21                 # draw\_cones na ax\_cones

22                 vis.draw\_cones(ax\_cones, fig, dists)

23

24                 # live-plot na ax\_live

25                 #plotter.val = names["distance sensor left front side"]

26                 #plotter.update(0)

27

28                 # automaty parkowania

29                 yaw = imu.getRollPitchYaw()[2] - yaw\_init

30                 parker.update\_state(names, yaw)

31                 if parker.state == "waiting\_for\_park":

32                     odom, spot = parker.update\_state(names, yaw)

33                     parker.exec\_path(odom, spot, names["distance sensor front left side"])

34                 #fig.canvas.draw\_idle()

35                 #fig.canvas.flush\_events()

36

37             # co IMAGE\_INTERVAL – przetwarzanie obrazów

38             if now - last\_image\_time >= IMAGE\_INTERVAL:

39                 last\_image\_time = now

40                 images = [get\_camera\_image(c) for c in cameras]

41                 names\_images = dict(zip(camera\_names, images))

42

43                 viss = vis.alt\_collect\_homo(names\_images, homographies, car, streams)

44                 #cv2.imwrite("img3\_vis.png",viss)

45             cv2.waitKey(1)

46

47         if now - last\_key\_time >= KEYBOARD\_INTERVAL:

48             last\_key\_time = now

49             check\_keyboard()