Komunikacja i sterowanie dronem za pośrednictwem MATLABa

Jakub Szczygieł

Skrypty inicjalizujące

ros_setup.m

```
1  rosinit('http://192.168.0.58:11311/');
2
3  takeoff_pub = rospublisher('/bebop/takeoff', 'std_msgs/Empty');
4  land_pub = rospublisher('/bebop/land', 'std_msgs/Empty');
5  emergency_pub = rospublisher('/bebop/emergency', 'std_msgs/Empty');
6  cmd_vel_pub = rospublisher('bebop/cmd_vel', 'geometry_msgs/Twist');
7
8  odometry_sub = rossubscriber('/bebop/odom', 'nav_msgs/Odometry');
9  image_raw_sub = rossubscriber('/bebop/image_raw', 'sensor_msgs/Image');
```

ros_connect.m

```
1  function ros_connect(ip, port)
2      Master_URI = strcat('http://', ip, ':', int2str(port), '/');
3      rosinit(Master_URI);
4  end
```

Skrypty publikujące

takeoff.m

```
function takeoff(takeoff_pub)
    msg = rosmessage('std_msgs/Empty');
    send(takeoff_pub, msg);
end
```

set_cmd_vel.m

```
function set_cmd_vel(cmd_vel_pub, cmd_vel)
    x = cmd_vel(1);
    y = cmd_vel(2);
    z = cmd_vel(3);
    omega = cmd_vel(4);
    msg = rosmessage('geometry_msgs/Twist');
    msg.Linear.X = x; % do przodu
    msg.Linear.Y = y; % w lewo
    msg.Linear.Z = z; % w górę
    msg.Angular.Z = omega; % obrót w lewo
    send(cmd_vel_pub, msg)
end
```

Skrypty subskrybujące

get_odometry.m

```
function odometry = get_odometry(odometry_sub)
    odom = receive(odometry_sub, 5); % czas oczekiwania [s]
    % pozycja
    pos = [0 0 0];
    pos(1) = odom.Pose.Pose.Position.X;
    pos(2) = odom.Pose.Pose.Position.Y;
    pos(3) = odom.Pose.Pose.Position.Z;
    % orientacja
    w = odom.Pose.Pose.Orientation.W;
    x = odom.Pose.Pose.Orientation.X;
    y = odom.Pose.Pose.Orientation.Y;
    z = odom.Pose.Pose.Orientation.Z;
    rot = quat2eul([w x y z]);

    odometry = [pos, rot];
end
```

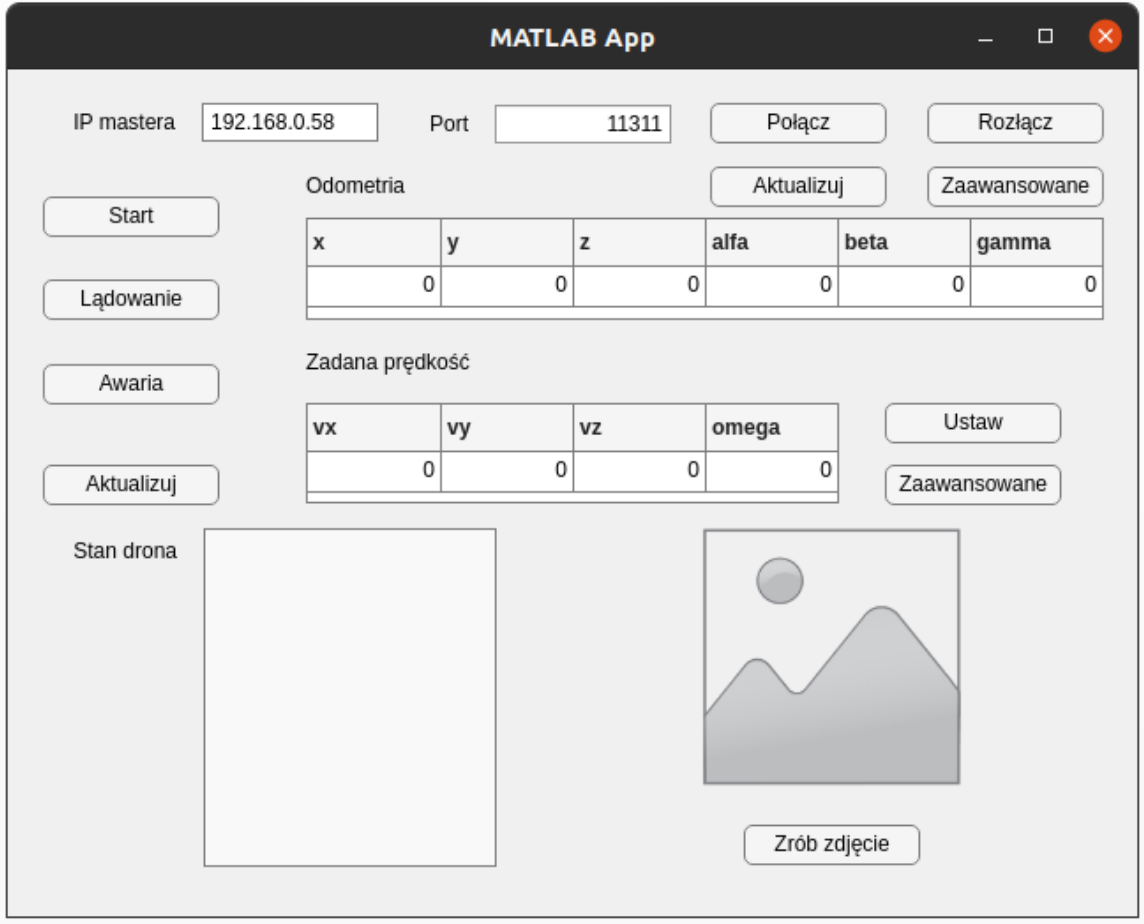
get_battery_state.m

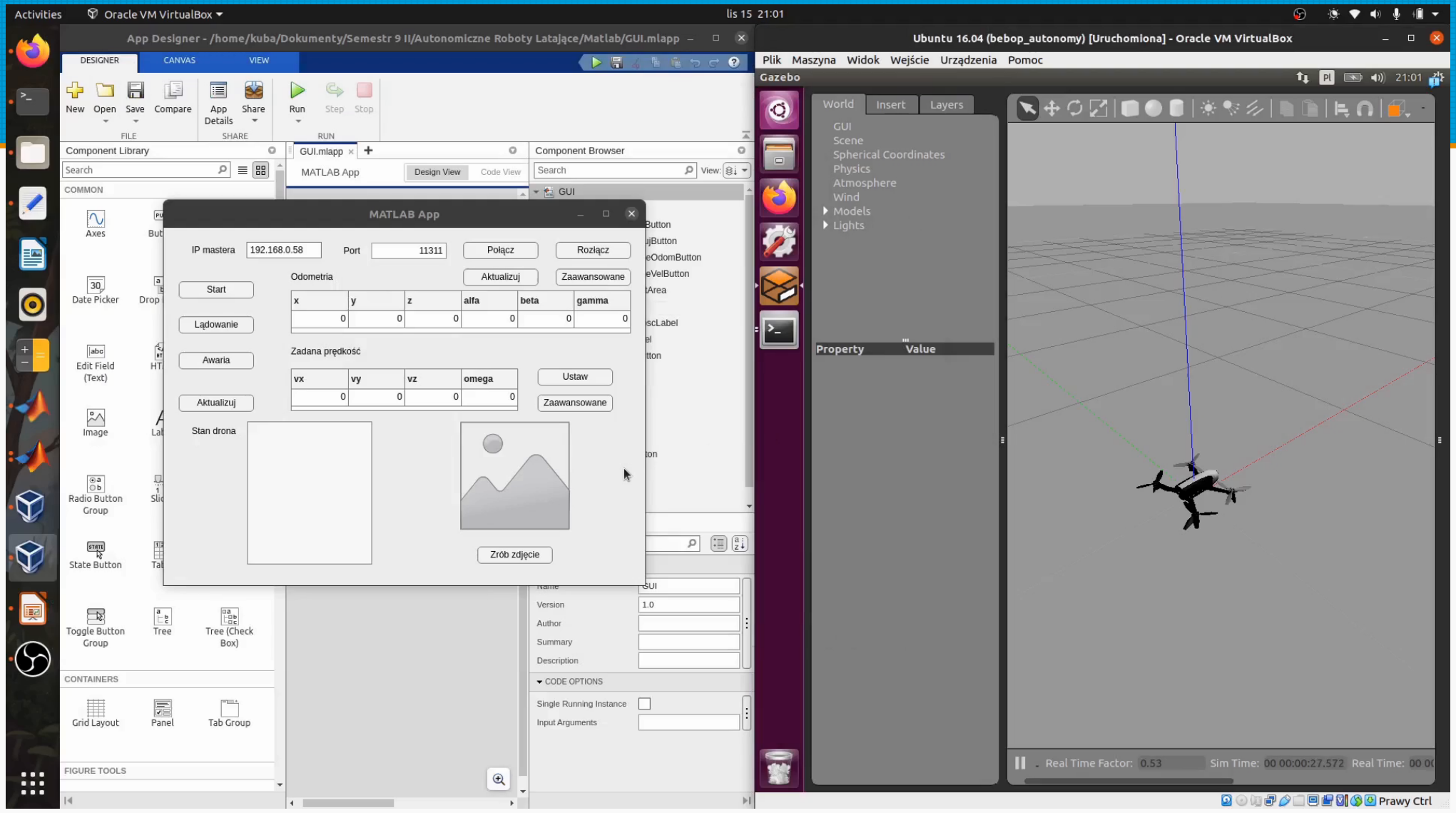
```
function battery_state = get_battery_state(battery_state_sub)
    battery_state = receive(battery_state_sub, 5).percent; % czas oczekiwania [s]
end

% procent naładowania baterii
```

get_image_raw.m

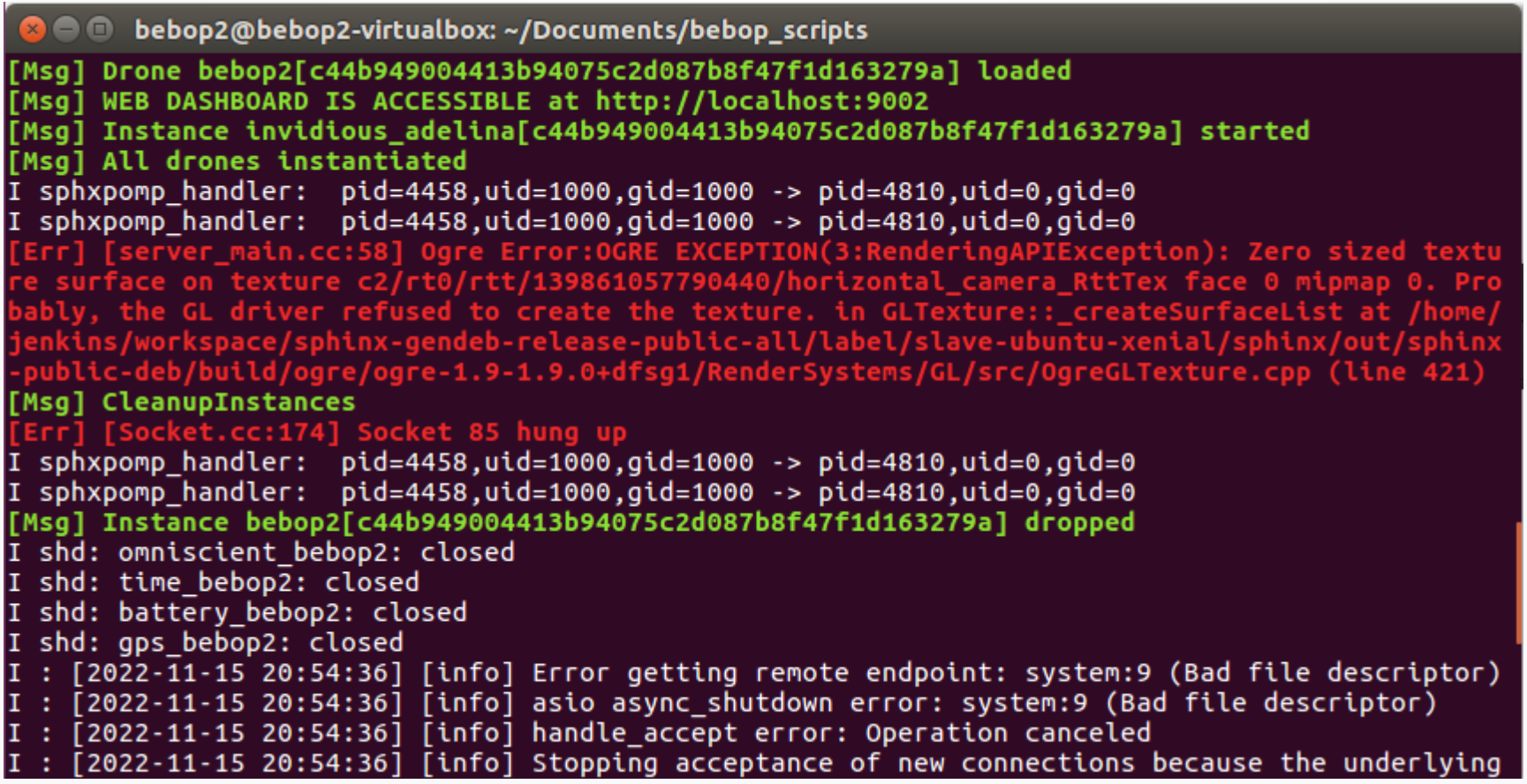
```
function img = get_image_raw(image_raw_sub)
    img_msg = receive(image_raw_sub, 5); % czas oczekiwania [s]
    img = readImage(img_msg);
end
```

Interfejs graficzny




Activities
Oracle VM VirtualBox
lis 15 21:01
App Designer - /home/kuba/Dokumenty/Semestr 9 II/Autonomiczne Roboty Latające/Matlab/GUI.mlapp
DESIGNER
CANVAS
VIEW
New
Open
Save
Compare
App Details
Share
Run
Step
Stop
FILE
SHARE
RUN
Component Library
Search
COMMON
Axes
Date Picker
Edit Field (Text)
Image
Radio Button Group
State Button
Toggle Button Group
Tree
Tree (Check Box)
CONTAINERS
Grid Layout
Panel
Tab Group
FIGURE TOOLS
GUI.mlapp
MATLAB App
Design View
Code View
Component Browser
GUI
Version
1.0
Author
Summary
Description
CODE OPTIONS
Single Running Instance
Input Arguments
MATLAB App
IP mastera
192.168.0.58
Port
11311
Połącz
Rozłącz
Start
Odometria
Aktualizuj
Zaawansowane
x
y
z
alfa
beta
gamma
Lądowanie
Awaria
Zadana prędkość
vx
vy
vz
omega
Ustaw
Aktualizuj
Zaawansowane
Stan drona
Zrób zdjęcie
Ubuntu 16.04 (bebop_autonomy) [Uruchomiona] - Oracle VM VirtualBox
Plik
Maszyna
Widok
Wejście
Urządzenia
Pomoc
Gazebo
21:01
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property
Value
Real Time Factor: 0.53
Sim Time: 00 00:00:27.572
Real Time:
Prawy Ctrl

Problem z kamerą



Dziękuję za uwagę