Синхронный танец нескольких квадрокоптеров с управлением по ROS (https://github.com/victoruni/vp_ardrone2)

1. Описание

2. Установка ardrone_autonomy

Установка пакета
Переходим на страницу проекта https://github.com/AutonomyLab/ardrone_autonomy
Там описана установка пакета из исходников. Установка в рабочее пространство catkin
cd ~/catkin_ws/src
git clone https://github.com/AutonomyLab/ardrone_autonomy.git -b hydro-devel
cd ~/catkin_ws
catkin_make

Для запуска пакета
rosrun ardrone_autonomy ardrone_driver

или запуск командного файла в котором можно выставить настройки,
отличные от настроек по умолчанию

roslaunch ardrone_autonomy ardrone1.launch

3. Создание пакета в catkin_ws

Для создания пакета в catkin

cd ~/catkin_ws/src
catkin_create_pkg vp_ardrone3 std_msgs rospy roscpp ardrone_autonomy

Теперь необходимо изменить файлы package.xml и CmakeLists.txt
В package.xml заполняем номер версии, автора, сведения о лицензии

Зависимости пакета указаны в блоках <buildtool_depend>(зависимости при сборке пакета) и <run_depend>(зависимости при выполнении пакета)

```
<buildtool_depend>catkin</buildtool_depend>
  <build_depend>ardrone_autonomy</build_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>rospy</build_depend>
  <build_depend>std_msgs</build_depend>
  <run_depend>ardrone_autonomy</run_depend>
  <run_depend>roscpp</run_depend>
  <run_depend>rospy</run_depend>
  <run_depend>std_msgs</run_depend>
```

Кроме этого в проекте мы будем использовать сообщения пользовательского типа или пользовательские сервисы, для этого необходимо создать файл описания сообщения или

сервиса и внести изменения в файл pakage.xml для генерации исходного кода в ROS для этого типа сообщений.
* msg: файлы сообщений (msg) — это простые текстовые файлы, которые описывают поля ROS-сообщения. Они используются для генерации исходного кода для сообщений на разных языках.
* srv: srv файл описывает сервис. Он состоит из двух частей: запрос и ответ.

Файлы msg хранятся в директории msg каталога пакета, а srv файлы, соответственно, хранятся в директории srv.

Мы создадим сервис для запроса сценариев с компьютеров с которыми связаны квадрокоптеры ardrone.

roscd vp_ardrone2
mkdir srv
cd srv
touch VpArdroneGetScenario.srv

И записываем в него следующую информацию

```
string resp
---
string answer
```

В файле pakage.xml раскомментируем следующие строки

```
<build_depend>message_generation</build_depend>
<run_depend>message_runtime</run_depend>
```

Вносим изменения в файл CmakeLists.txt

здесь -

```
find_package(catkin REQUIRED COMPONENTS
  ardrone_autonomy
  roscpp
  rospy
  std_msgs
  message_generation
)
```

здесь -

```
catkin_package(
#  INCLUDE_DIRS include
#  LIBRARIES my_kinect_1
   CATKIN_DEPENDS roscpp rospy std_msgs message_runtime
#  DEPENDS system_lib
)
```

Добавляем

```
 add_service_files(
   FILES
   VpArdroneGetScenario.srv
 )
```

Раскомментируем эти строки

Копируем скрипты в папку nodes

Сборка и компиляция в catkin
cd ~/catkin_ws
catkin_make

4. Установка Rosbridge

Создан в Университет Брауна, rosbridge был первоначально предназначен для любого не ROS процессов клиента для работы с системой ROS. Rosbridge позволяет внешним клиентам иметь доступ к темам и сервисам ROS (публикация и получение из тем, вызов сервисов). Rosbridge является частью мете-пакета rosbridge_suite, включающего различные дополнительные пакеты для реализации протокола rosbridge.
Установка пакета
sudo apt-get install ros-hydro-rosbridge-server
sudo apt-get install ros-hydro-rosbridge-suite

Программа писалась год назад и веб-интерфейсе отображался вид с камеры. Поэтому устанавливался пакет mjpeg-server.

sudo apt-get install ros-hydro-mjpeg-server

К сожалению новые версии Google Chrome не поддерживают отображение потокового видео комп очень сильно зависает, поэтому из web-интерфейса убрана камера

5. Установка www

Будем использовать пакет roswww_pkg

cd ~/catkin_ws/src
git clone https://github.com/tork-a/roswww_pkg.git
cd ~/catkin_ws
catkin_make

6. Запуск

Один из ноутбуков использовался в качестве табло, где можно выбрать предварительно написанный сценарий для каждого квадрокоптера, выбрать квадрокоптеры для запуска, запустить шоу на выполнение, на табло также отображается некоторая информация с датчиков.
Здесь был дополнительно установлен пакет Rosbridge (о работе пакета rosbridge для Turtlebot я писал здесь)
При запуске сценариев на выполнение центральный комп, считывает пошагово команды сценария для каждого нетбука, и отправляет на каждый комп, по окончании выполнения команды от нетбука приходит сигнал, и центральный комп отправляет на выполнение следущую команду сценария.
Пример сценария

```
scenario1
start;5
#wait;5
takeoff;5.0
#;
#; эмуляция фигуры 13 подъем до 1.0-1.5-2.0-2.5 м
cmd_vel;3.0;0.0;0.0;0.3;0.0;1100;1;0;0;
cmd_vel;3.0;0.0;0.0;0.3;0.0;1700;1;0;0;
cmd_vel;3.0;0.0;0.0;0.3;0.0;2200;1;0;0;
#;
#; эмуляция фигуры 8 опускание до 0.5 м
userfly2;6.0;0.0;0.0;-0.2;1.0;750;2;0;1;
userfly1;4.0;0.0;0.0;0.0;0.2;3000;2;0;1;
#;
# анимации 16
#fly;6.0;15;0.0
#;
# круговое - эмуляция 14
userfly5;10.0;0.15;0.2;0.08;0.0;2000;1;0;1;
#;
# анимации 1, 2
fly;2.0;0;0.0
fly;2.0;1;0.0
#;
#; эмуляция фигуры 13 опускание до 1.5 м
cmd_vel;3.0;0.0;0.0;-0.2;0.0;1500;2;0;0;
#;
# анимации 15
#fly;6.0;14;0.0
#;
#;или эмуляция фигуры 13 подъем до 2.5 м
#cmd_vel;4.0;0.0;0.0;0.2;0.0;2200;1;0;0;
#;
#; фигуры 5,6
fly;4.0;4;0.0
fly;4.0;5;0.0
#;
#; дрожание 9
fly;3.0;8;0.0
cmd_vel;4.0;0.0;0.0;0.2;0.0;2200;1;0;0;#;
#; сальто
```

fly;5.0;16;0.0
#; опускание по высоте
cmd_vel;3.0;0.0;0.0;-0.2;0.0;1500;2;0;0;
cmd_vel;3.0;0.0;0.0;-0.2;0.0;1100;2;0;0;
cmd_vel;3.0;0.0;0.0;-0.2;0.0;650;2;0.0;0;
#;end
land;5.0
end

Или простой — запуск только светодиодных анимаций без взлета
ledanimation1
start;5
wait;5
led;5.0;1;1.0;4
led;5.0;0;1.0;4
led;5.0;2;1.0;4
led;5.0;3;1.0;4
led;5.0;4;1.0;4
led;5.0;5;1;4
led;5.0;6;1;4
led;5.0;7;1;4
led;5.0;8;1;4
led;5.0;9;1;4
led;5.0;10;1;4
led;5.0;11;1;4
led;5.0;12;1;4
led;5.0;13;1;4
end

На табло отображается текущая команда сценария для каждого квадрокоптера. И просто изображение с центральной камеры одного из квадрокоптеров.

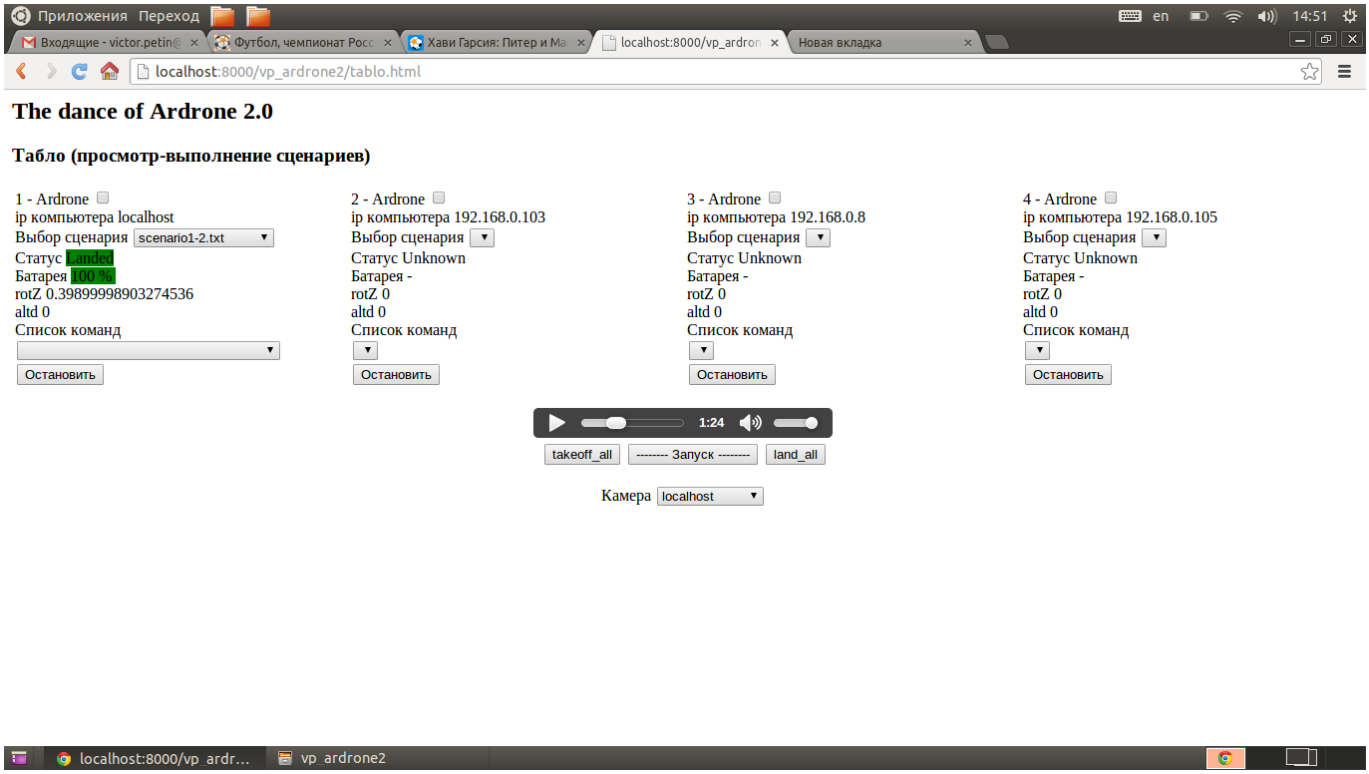
Запуск(для каждого нетбука и ноутбука): в качестве терминала удобно использовать GuakeTerminal
1 монитор
roscore

2 монитор
roslaunch vp_ardrone2 www.launch

3 монитор — после соединения нетбука с квадрокоптером
roslaunch ardrone_autonomy ardrone1.launch
(roslaunch ardrone_autonomy ardrone2.launch)

В браузере ноутбука (только Google Chrome (не Chrome!!!) — в других нет запуска js файла музыкального сопровождения)

Браузер:
localhost:8000/vp_ardrone2/tablo.html


Приложения Переход
en 14:51
Входящие - victor.petin@
Футбол, чемпионат Росс
Хави Гарсия: Питер и Ма
localhost:8000/vp_ardron
Новая вкладка
localhost:8000/vp_ardrone2/tablo.html
The dance of Ardrone 2.0
Табло (просмотр-выполнение сценариев)
1 - Ardrone
ip компьютера localhost
Выбор сценария scenario1-2.txt
Статус landed
Батарея 100 %
rotZ 0.39899998903274536
altd 0
Список команд
Остановить
2 - Ardrone
ip компьютера 192.168.0.103
Выбор сценария
Статус Unknown
Батарея -
rotZ 0
altd 0
Список команд
Остановить
3 - Ardrone
ip компьютера 192.168.0.8
Выбор сценария
Статус Unknown
Батарея -
rotZ 0
altd 0
Список команд
Остановить
4 - Ardrone
ip компьютера 192.168.0.105
Выбор сценария
Статус Unknown
Батарея -
rotZ 0
altd 0
Список команд
Остановить
1:24
takeoff_all
-------- Запуск --------
land_all
Камера localhost
localhost:8000/vp_ardr...
vp_ardrone2