ШПАРГАЛКА ROS - HYDRO v1.0

ROS.org

РАБОЧИЕ ПРОСТРАНТСВА

Создать рабочее пространство

```
mkdir <директория> && cd <директория>
wstool init src
catkin_make
source devel/setup.bash
```

Добавить репозиторий в рабочее пространство

```
roscd; cd ../src
wstool set repo_name --git http://github.com/org/repo_name.
git --version=hydro-devel
rstool up
```

Решить зависимости в рабочем пространстве

```
sudo rosdep init  # единожды, во время установки ROS
rosdep update
rosdep install --from-paths src --ignore-src \
  --rosdistro=hydro -y
```

ПАКЕТЫ

Создать пакет

```
catkin_create_pkg <имя пакета> [зависимости ...]
```

Структура каталогов в пакете

include/package_name	заголовочные файлы C++
src	исходные коды, библиотеки Python в подкаталогах
scripts	Python-сценарии и узлы
msg, srv, action	сообщения, службы и действия

Публикация пакета в репозитории

```
catkin_generate_changelog
catkin_prepare_release
bloom-release --track hydro --ros-distro hydro <репозиторий>
```

Следует помнить

- Тестируемая логика
- Вывод в отдельные топики диагностической информации
- Вынесение системных зависимостей в отдельный пакет

CMakeLists.txt

Скелет

```
cmake_minimum_required(VERSION 2.8.3)
project(<имя пакета>)
find_package(catkin REQUIRED)
catkin_package()
```

Зависимости пакета

Чтобы использовать заголовки и библитоеки другого пакета, или Описанные в нем макросы CMake, укажите build-time-зависимость:

```
find_package(catkin REQUIRED COMPONENTS roscpp)
```

Указать зависимым пакетам, какие компоненты являются «глобально видимыми», доступными для использования извне:

```
catkin_package(
  INCLUDE_DIRS include
  LIBRARIES ${PROJECT_NAME}
  CATKIN_DEPENDS roscpp)
```

Сообщения, службы

Располагаются после find_package(), но перед catkin_package(). Пример:

```
find_package(catkin REQUIRED
  COMPONENTS message_generation std_msgs)
add_message_files(FILES MyMessage.msg)
add_service_files(FILES MyService.msg)
generate_messages(DEPENDENCIES std_msgs)
catkin_package(
  CATKIN_DEPENDS message_runtime std_msgs)
```

Сборка библиотек, исполняемых файлов

Размещается после вызова catkin_package().

```
add_library(${PROJECT_NAME} src/main)
add_executable(${PROJECT_NAME}_node src/main)
target_link_libraries(
  ${PROJECT_NAME}_node ${catkin_LIBRARIES})
```

Установка пакета

```
install(TARGETS ${PROJECT_NAME}
  DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION})
install(TARGETS ${PROJECT_NAME}_node
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})install(PRO-
GRAMS scripts/myscript
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})
install(DIRECTORY launch
  DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION})
```

РАБОТА СИСТЕМЫ

Запустить мастер-сервер ROS можно отдельно:

```
roscore
```

Иначе, roslaunch запустит собственный roscore автоматически, если не найдет уже запущенный:

```
roslaunch <имя пакета> <конфигурация запуска>.launch
```

При выполнении с флагом `--wait`, roslaunch будет ждать внешний roscore.

Узлы, топики, сообщения

```
rosnode list
rostopic list
rostopic echo cmd_vel
rostopic hz cmd_vel
rostopic info cmd_vel
rosmsg show geometry_msgs/Twist
```

Удаленное подключение

Переменные окружения для мастер-сервера ROS:

- `ROS_IP` или `ROS_HOSTNAME` : доступный клиентам адрес мастер-сервера.
- `ROS_MASTER_URI` : URL, содержащий этот адрес.

Переменные окружения удаленных клиентов:

- `ROS_IP` или `ROS_HOSTNAME` : адрес клиента, доступный мастер-серверу.
- `ROS_MASTER_URI` : URL мастер-сервера.

Для отладки, проверьте результаты rosnode ping для разных узлов, воспользуйтесь диагностической утилитой roswtf на обеих сторонах

Консоль ROS

Установите rqt_logger_level и используйте rqt_console для мониторинга. Для автоматической загрузки параметров, создайте конфигурационный файл и добавьте в него строку для своего пакета:

```
log4j.logger.ros.<имя пакета>=DEBUG
```

После этого добавьте переменные в сессию:

```
export ROSCONSOLE_CONFIG_FILE=<путь до файла конфигурации>
```

Используйте roslaunch с флагом --screen чтобы выводить сообщения всех узлов на экран. Каждый <node> в launch-файле имеет параметр `output="screen"` для выбора устройства вывода.

CLEARPATH ROBOTICS™

www.clearpathrobotics.com/resources/technical-documentation © 2014 Clearpath Robotics, Inc. All Rights Reserved