

# 2015 ROS Yaz Okulu Uygulamalar - I

Revizyon : 1.00  
Tarih : 12.08.2015



## DOKÜMAN REVİZYON SAYFASI

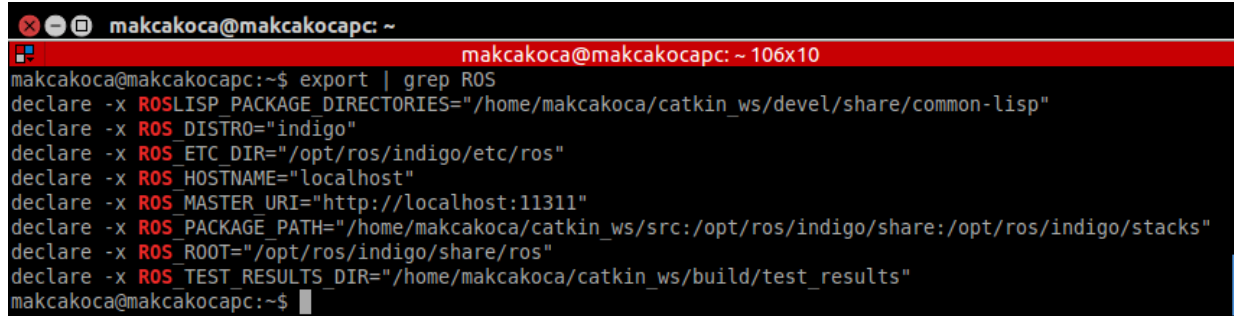
REV. NO	TARİH	SAYFA NO.	AÇIKLAMA
1.00	-	Tümü	İlk versiyon

## İÇİNDEKİLER

<b>1.Uygulama-1 : ROS Ortamının Hazırlanması.....</b>	<b>1</b>
<b>2.Uygulama-2 : Catkin Paket Oluşturma.....</b>	<b>2</b>
<b>3.Uygulama-3 : TurtleSim Uygulaması.....</b>	<b>5</b>
<b>4.Uygulama-4 : Message Oluşturma.....</b>	<b>10</b>
<b>5.Uygulama-5 : Publisher &amp; Subscriber Uygulaması.....</b>	<b>12</b>
5.1Publisher Yapımı.....	12
5.2Subscriber Yapımı.....	13
5.3Derleme.....	14
5.4Çalıştırma.....	14
<b>6.Uygulama-6 : Service &amp; Client Uygulaması.....</b>	<b>15</b>
6.1srv oluşturma.....	15
6.2Server Oluşturma.....	15
6.3Client Oluşturma.....	16
6.4Derleme.....	17
6.5Uygulamayı Çalıştırma.....	18

## 1. Uygulama-1 : ROS Ortamının Hazırlanması

```
$ export | grep ROS
```



```
makcakoca@makcakocapc: ~  
makcakoca@makcakocapc: ~ 106x10  
makcakoca@makcakocapc:~$ export | grep ROS  
declare -x ROSLISP_PACKAGE_DIRECTORIES="/home/makcakoca/catkin_ws/devel/share/common-lisp"  
declare -x ROS_DISTRO="indigo"  
declare -x ROS_ETC_DIR="/opt/ros/indigo/etc/ros"  
declare -x ROS_HOSTNAME="localhost"  
declare -x ROS_MASTER_URI="http://localhost:11311"  
declare -x ROS_PACKAGE_PATH="/home/makcakoca/catkin_ws/src:/opt/ros/indigo/share:/opt/ros/indigo/stacks"  
declare -x ROS_ROOT="/opt/ros/indigo/share/ros"  
declare -x ROS_TEST_RESULTS_DIR="/home/makcakoca/catkin_ws/build/test_results"  
makcakoca@makcakocapc:~$
```

```
$ gedit ~/.bashrc
```

Gedit editöründe açılan ekrana `<source /opt/ros/indigo/setup.bash>` ve `<source /home/<kullanıcı_adı>/catkin_ws/devel/setup.bash >` kodu eklenir.

```
$ mkdir -p ~/catkin_ws/src
```

```
$ cd ~/catkin_ws/src
```

```
$ catkin_init_workspace
```

```
$ cd ~/catkin_ws/
```

```
$ catkin_make
```

```
$ source devel/setup.bash
```

bashrc'de yapılan değişiklikler önceden açılan terminalde algılanabilmesi için bash komutu ile terminal yenilenmektedir.

## 2. Uygulama-2 : Catkin Paket Oluşturma

```
$ cd ~/catkin_ws/src
```

```
$ catkin_create_pkg beginner_tutorials std_msgs roscpp
```

```
makcakoca@makcakocapc: ~/catkin_ws/src$ catkin_create_pkg beginner_tutorials std_msgs roscpp
Created file beginner_tutorials/CMakeLists.txt
Created file beginner_tutorials/package.xml
Created folder beginner_tutorials/include/beginner_tutorials
Created folder beginner_tutorials/src
Successfully created files in /home/makcakoca/catkin_ws/src/beginner_tutorials.
Please adjust the values in package.xml.
makcakoca@makcakocapc:~/catkin_ws/src$
```

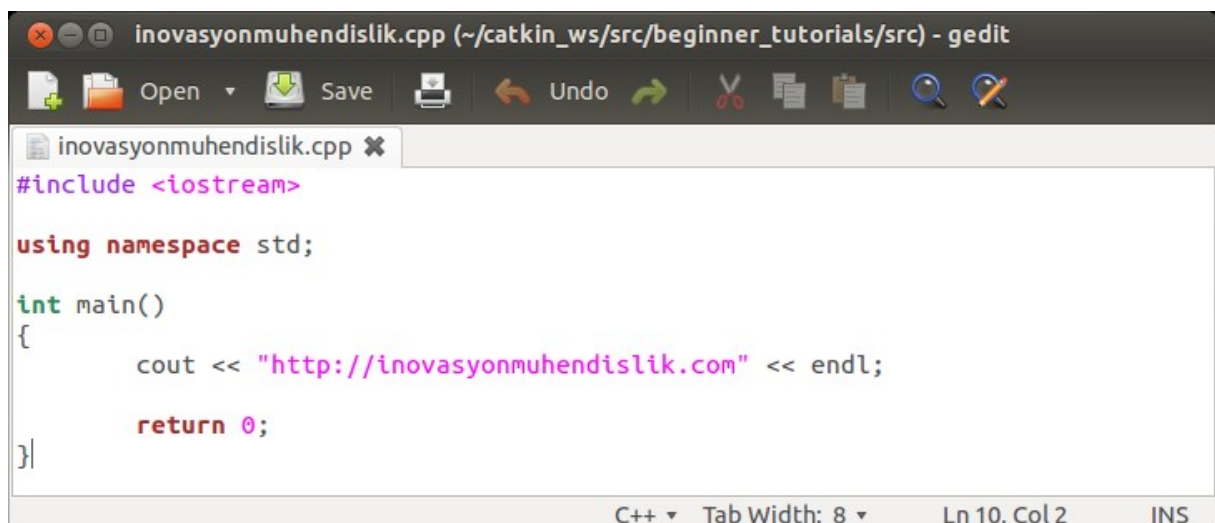
```
$ cd beginner_tutorials
```

```
$ ls
```

```
Please adjust the values in package.xml.
makcakoca@makcakocapc:~/catkin_ws/src$ cd beginner_tutorials/
makcakoca@makcakocapc:~/catkin_ws/src/beginner_tutorials$ ls
CMakeLists.txt  include  package.xml  src
makcakoca@makcakocapc:~/catkin_ws/src/beginner_tutorials$
```

```
$ cd src
```

```
$ gedit inovasyonmuhendislik.cpp
```



```
inovasyonmuhendislik.cpp (~/catkin_ws/src/beginner_tutorials/src) - gedit
Open Save Undo Cut Copy Paste Find
inovasyonmuhendislik.cpp
#include <iostream>

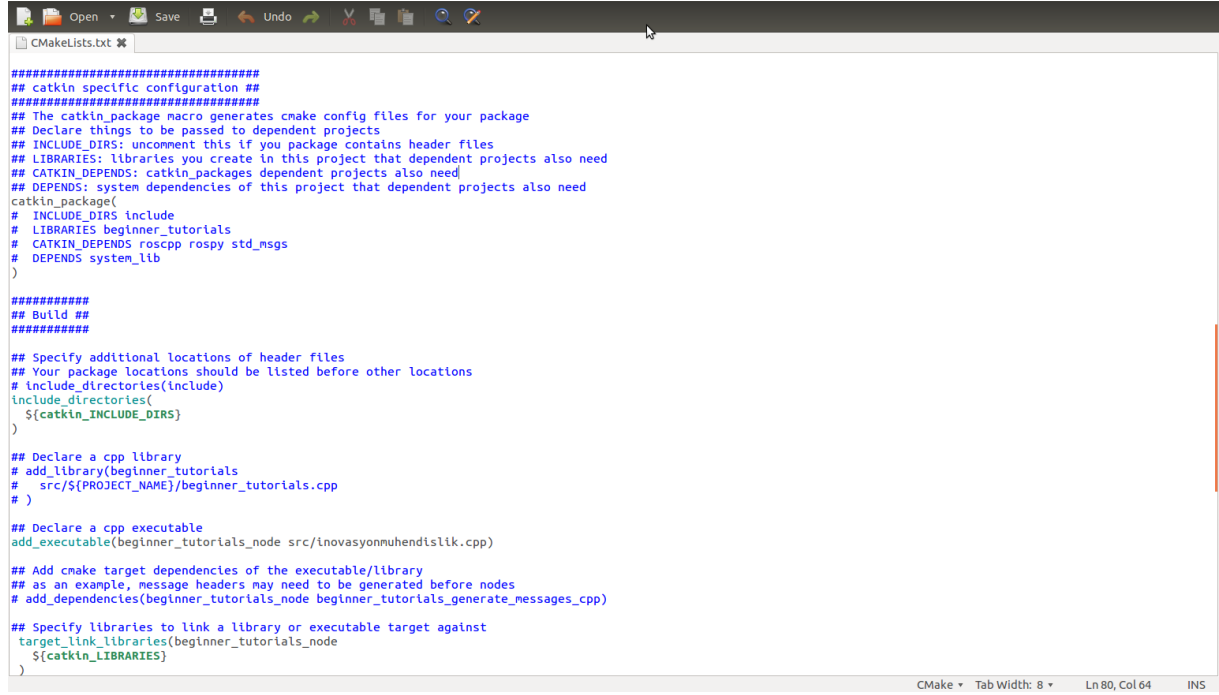
using namespace std;

int main()
{
    cout << "http://inovasyonmuhendislik.com" << endl;

    return 0;
}
```

```
$ cd ..
```

```
$ gedit CMakeLists.txt
```



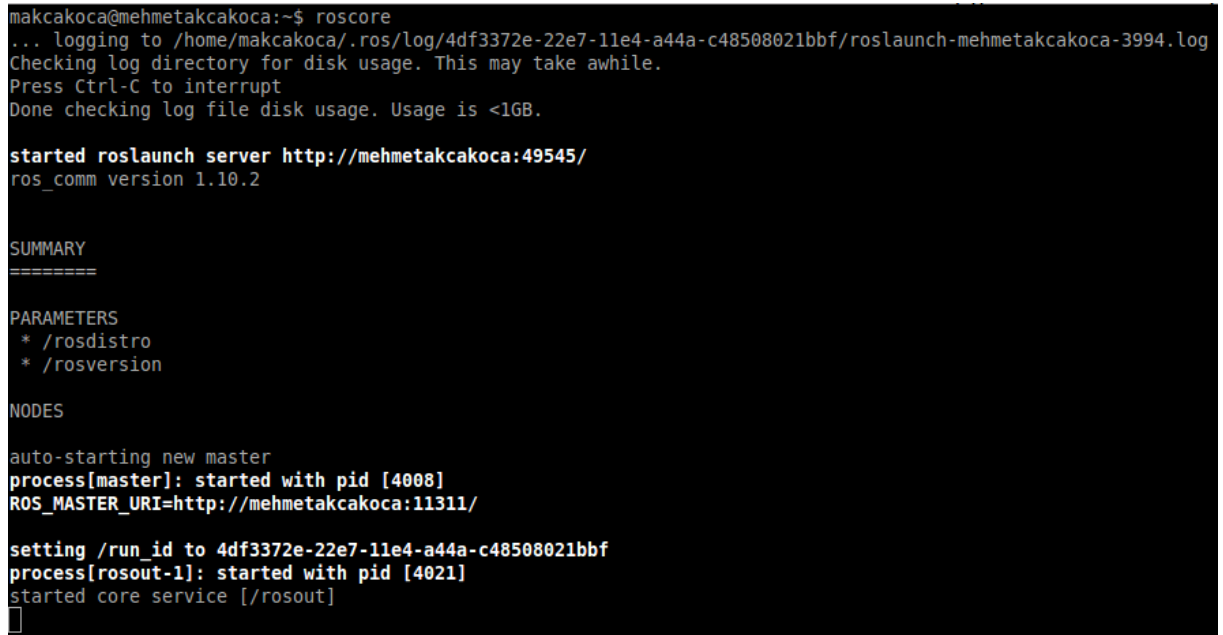
```
#####  
## catkin specific configuration ##  
#####  
## The catkin_package macro generates cmake config files for your package  
## Declare things to be passed to dependent projects  
## INCLUDE_DIRS: uncomment this if you package contains header files  
## LIBRARIES: libraries you create in this project that dependent projects also need  
## CATKIN_DEPENDS: catkin_packages dependent projects also need  
## DEPENDS: system dependencies of this project that dependent projects also need  
catkin_package(  
  # INCLUDE_DIRS include  
  # LIBRARIES beginner_tutorials  
  # CATKIN_DEPENDS roscpp rospy std_msgs  
  # DEPENDS system_lib  
)  
  
#####  
## Build ##  
#####  
  
## Specify additional locations of header files  
## Your package locations should be listed before other locations  
include_directories(include)  
include_directories(  
  ${catkin_INCLUDE_DIRS}  
)  
  
## Declare a cpp library  
# add_library(beginner_tutorials  
#   src/${PROJECT_NAME}/beginner_tutorials.cpp  
# )  
  
## Declare a cpp executable  
add_executable(beginner_tutorials_node src/inovasyonmuhendislik.cpp)  
  
## Add cmake target dependencies of the executable/library  
## as an example, message headers may need to be generated before nodes  
# add_dependencies(beginner_tutorials_node beginner_tutorials_generate_messages_cpp)  
  
## Specify libraries to link a library or executable target against  
target_link_libraries(beginner_tutorials_node  
  ${catkin_LIBRARIES}  
)
```

```
$ cd ~/catkin_ws/
```

```
$ catkin_make
```

Yazılana kodun çalıştırılması için iki terminal açılır. İlk terminalde roscore çalıştırılır.

```
$ roscore
```



```
makcakoca@mehmetakcakoca:~$ roscore  
... logging to /home/makcakoca/.ros/log/4df3372e-22e7-11e4-a44a-c48508021bbf/roslaunch-mehmetakcakoca-3994.log  
Checking log directory for disk usage. This may take awhile.  
Press Ctrl-C to interrupt  
Done checking log file disk usage. Usage is <1GB.  
  
started roslaunch server http://mehmetakcakoca:49545/  
ros_comm version 1.10.2  
  
SUMMARY  
=====  
  
PARAMETERS  
* /roscout  
* /rosversion  
  
NODES  
  
auto-starting new master  
process[master]: started with pid [4008]  
ROS_MASTER_URI=http://mehmetakcakoca:11311/  
  
setting /run_id to 4df3372e-22e7-11e4-a44a-c48508021bbf  
process[roscout-1]: started with pid [4021]  
started core service [/roscout]  
[ ]
```

Diğer terminalde ise oluşturulan ros paketi çalıştırılır.

```
$ rosrn beginner_tutorials beginner_tutorials_node
```

```
makcakoca@mehmetakcakoca:~$ rosrn beginner_tutorials beginner_tutorials_node  
http://inovasyonmuhendislik.com  
makcakoca@mehmetakcakoca:~$
```

### 3. Uygulama-3 : TurtleSim Uygulaması

Öncelikle [wiki.ros.org](http://wiki.ros.org)'da bulunan derslerin ros paketleri indirilir.

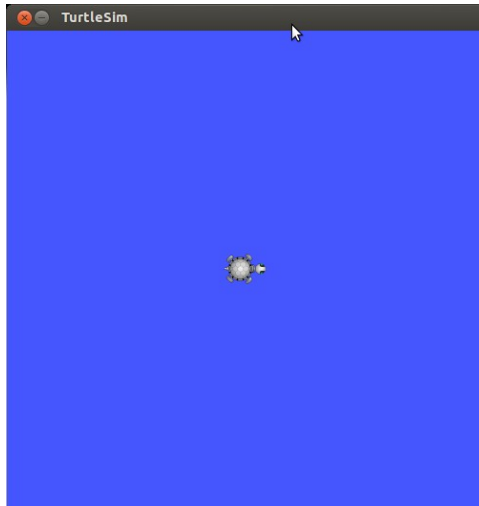
```
$ sudo apt-get install ros-indigo-ros-tutorials
```

#### Terminal-1

```
$ roscore
```

#### Terminal-2

```
$ rosrunc turtlesim turtlesim_node
```



#### Terminal-3

```
$ rosrunc turtlesim turtle_teleop_key
```

```
makcakoca@mehmetakcakoca:~$ rosrunc turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle.
█
```





#### Terminal-4

```
$ rosnodet list
```

```
makcakoca@mehtetakcakoca:~$ rosnodet list
/rosout
/teleop_turtle
/turtlesim
```

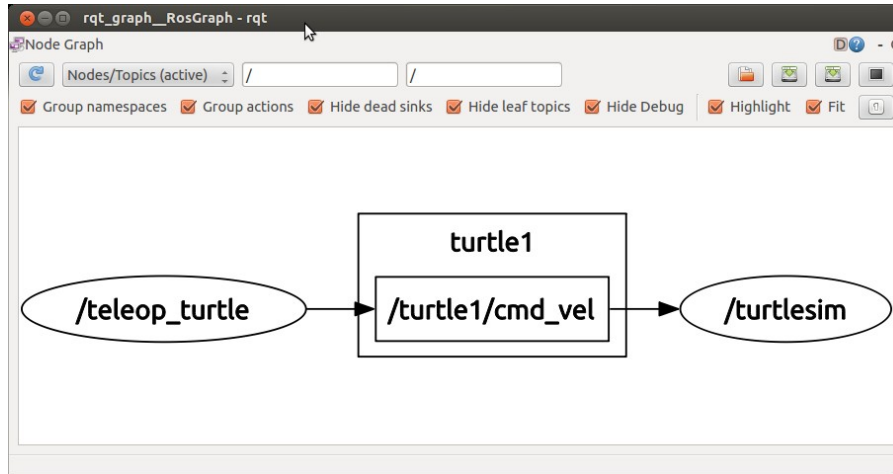
```
$ rostopic list
```

```
makcakoca@mehtetakcakoca:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
```

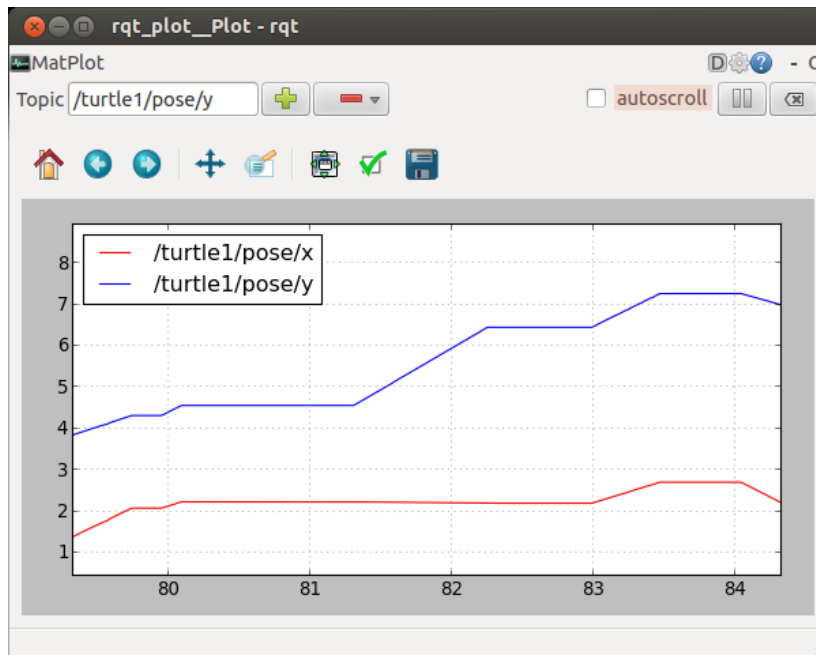
```
$ rostopic echo /turtle1/cmd_vel
```

```
makcakoca@mehtetakcakoca:~$ rostopic echo /turtle1/cmd_vel
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```

```
$ rqt_graph
```



```
$ rqt_plot
```



```
$ rostopic info /turtle1/cmd_vel
```

```

makcakoca@mehmetakcakoca:~$ rostopic info /turtle1/cmd_vel
Type: geometry_msgs/Twist

Publishers:
 * /teleop_turtle (http://mehmetakcakoca:51757/)

Subscribers:
 * /turtlesim (http://mehmetakcakoca:57532/)

```

```
$ rosmg show geometry_msgs/Twist
```

```
makcakoca@mehmetakcakoca:~$ rosmg show geometry_msgs/Twist
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z
```

```
$ rosservice list
```

```
makcakoca@mehmetakcakoca:~$ rosservice list
/clear
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/spawn
/teleop_turtle/get_loggers
/teleop_turtle/set_logger_level
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/get_loggers
/turtlesim/set_logger_level
```

```
$ rosservice type /spawn
```

```
makcakoca@mehmetakcakoca:~$ rosservice type /spawn
turtlesim/Spawn
```

```
$ rossrv show turtlesim/Spawn
```

```
makcakoca@mehmetakcakoca:~$ rossrv show turtlesim/Spawn
float32 x
float32 y
float32 theta
string name
---
string name
```

```
$ rosservice call /spawn 3 2 0 im_tosbaa
```



```
$ rosparam list
```

```
makcakoca@mehmetakcakoca:~$ rosparam list
/background_b
/background_g
/background_r
/rosdistro
/roslaunch/uris/host_mehmetakcakoca__55530
/rosversion
/run_id
```

```
$ rosparam get /background_b
```

```
makcakoca@mehmetakcakoca:~$ rosparam get /background_b
255
```

```
$ rosparam set /background_b 10
```

```
makcakoca@mehmetakcakoca:~$ rosparam set /background_b 10
```

```
$ rosparam get /background_b
```

```
makcakoca@mehmetakcakoca:~$ rosparam get /background_b
10
```

#### 4. Uygulama-4 : Message Oluşturma

```
$ cd ~/catkin_ws/src/beginner_tutorials  
$ mkdir msg  
$ echo "int64 num" > msg/Num.msg
```

ya da Num.msg içerişi aşağıdaki gibi doldurulur.

```
string first_name  
string last_name  
uint8 age  
uint32 score
```

```
$ gedit package.xml
```

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

```
$ gedit CMakeLists.txt
```

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

```
catkin_package(  
  ...  
  CATKIN_DEPENDS message_runtime ...  
  ...)
```

```
add_message_files(  
  FILES  
  Num.msg  
)
```

```
generate_messages(  
  DEPENDENCIES  
    std_msgs  
)
```

```
$ cd ~/catkin_ws  
$ catkin_make
```

```
$ rosmmsg show beginner_tutorials/Num
```

## 5. Uygulama-5 : Publisher & Subscriber Uygulaması

### 5.1 Publisher Yapımı

```
$ cd ~/catkin_ws/src/beginner_tutorials/src  
$ gedit talker.cpp
```

```
#include "ros/ros.h"  
#include "std_msgs/String.h"  
  
#include <sstream>  
  
int main(int argc, char **argv)  
{  
    ros::init(argc, argv, "talker");  
  
    ros::NodeHandle n;  
  
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);  
  
    ros::Rate loop_rate(10);  
  
    int count = 0;  
    while (ros::ok())  
    {  
        std_msgs::String msg;  
  
        std::stringstream ss;  
        ss << "hello world " << count;  
        msg.data = ss.str();  
  
        ROS_INFO("%s", msg.data.c_str());  
        chatter_pub.publish(msg);  
  
        ros::spinOnce();  
        loop_rate.sleep();  
        ++count;  
    }  
    return 0;  
}
```

```
$ cd ~/catkin_ws/src/beginner_tutorials
$ gedit CMakeLists.txt
```

```
add_executable(talker src/talker.cpp)
target_link_libraries(talker
  ${catkin_LIBRARIES}
)
```

## 5.2 Subscriber Yapımı

```
$ cd ~/catkin_ws/src/beginner_tutorials/src
$ gedit listener.cpp
```

```
#include "ros/ros.h"
#include "std_msgs/String.h"

void chatterCallback(const std_msgs::String::ConstPtr& msg)
{
  ROS_INFO("I heard: [%s]", msg->data.c_str());
}

int main(int argc, char **argv)
{
  ros::init(argc, argv, "listener");

  ros::NodeHandle n;

  ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);

  ros::spin();

  return 0;
}
```





```
$ cd ~/catkin_ws/src/beginner_tutorials  
$ gedit CMakeLists.txt
```

```
add_executable(listener src/listener.cpp)  
target_link_libraries(listener  
  ${catkin_LIBRARIES}  
)
```

### 5.3 Derleme

```
$ cd ~/catkin_ws/  
$ catkin_make
```

### 5.4 Çalıştırma

#### Terminal-1:

```
$ roscore
```

#### Terminal-2:

```
$ rosrun beginner_tutorials talker
```

#### Terminal-3:

```
$ rosrun beginner_tutorials listener
```

## 6. Uygulama-6 : Service & Client Uygulaması

### 6.1 srv oluşturma

```
$ roscd beginner_tutorials
$ mkdir srv
$ cd srv
$ gedit AddTwoInts.srv
```

```
int64 a
int64 b
---
int64 sum
```

```
$ roscd beginner_tutorials
$ gedit CMakeLists.txt
```

```
add_service_files(
  FILES
  AddTwoInts.srv
)
```

```
$ cd ~/catkin_ws
$ catkin_make
```

### 6.2 Server Oluşturma

```
$ cd ~/catkin_ws/src/beginner_tutorials/src
$ gedit add_two_ints_server.cpp
```

```
#include "ros/ros.h"
#include "beginner_tutorials/AddTwoInts.h"

bool add(beginner_tutorials::AddTwoInts::Request &req,
         beginner_tutorials::AddTwoInts::Response &res)
{
    res.sum = req.a + req.b;
    ROS_INFO("request: x=%ld, y=%ld", (long int)req.a, (long int)req.b);
    ROS_INFO("sending back response: [%ld]", (long int)res.sum);
}
```

```
return true;
}

int main(int argc, char **argv)
{
    ros::init(argc, argv, "add_two_ints_server");
    ros::NodeHandle n;

    ros::ServiceServer service = n.advertiseService("add_two_ints", add);
    ROS_INFO("Ready to add two ints.");
    ros::spin();

    return 0;
}
```

### 6.3 Client Oluşturma

```
$ cd ~/catkin_ws/src/beginner_tutorials/src
$ gedit add_two_ints_client.cpp
```

```
#include "ros/ros.h"
#include "beginner_tutorials/AddTwoInts.h"
#include <cstdlib>

int main(int argc, char **argv)
{
    ros::init(argc, argv, "add_two_ints_client");
    if (argc != 3)
    {
        ROS_INFO("usage: add_two_ints_client X Y");
        return 1;
    }

    ros::NodeHandle n;
    ros::ServiceClient client = n.serviceClient<beginner_tutorials::AddTwoInts>("add_two_ints");
    beginner_tutorials::AddTwoInts srv;
    srv.request.a = atoll(argv[1]);
    srv.request.b = atoll(argv[2]);
```

```
if (client.call(srv))
{
    ROS_INFO("Sum: %ld", (long int)srv.response.sum);
}
else
{
    ROS_ERROR("Failed to call service add_two_ints");
    return 1;
}

return 0;
}
```

## 6.4 Derleme

```
$ cd ~/catkin_ws/src/beginner_tutorials/
$ gedit CMakeLists.txt
```

```
add_executable(server src/add_two_ints_server.cpp)
target_link_libraries(server
    ${catkin_LIBRARIES}
)

add_executable(client src/add_two_ints_client.cpp)
target_link_libraries(client
    ${catkin_LIBRARIES}
)
```

```
$ cd ~/catkin_ws
$ catkin_make
```



## 6.5 Uygulamayı Çalıştırma

### Terminal-1:

```
$ roscore
```

### Terminal-2:

```
$ rosrn beginner_tutorials server
```

### Terminal-3:

```
$ rosrn beginner_tutorials client 2 3
```