

Baxter Nasıl Kullanılır?

1)Açılım ve RobotEnable Kodu:

Baxter'ı açmak için arka sol kısımdaki düğmeye basınız Şekil 1.



Şekil 1

Açıldıktan sonra terminali açmak için ctrl+alt+f1 kısayolunu kullanabilirsiniz. Giriş ekranı karşınıza çıkınca kullanıcı adı: ruser ve şifre: rethink bilgileri ile giriş yapınız. Bu işlemi baxter'a klavye bağlayarak robotun bilgisayarından yapabileceğiniz gibi ssh ile bağlanarak da yapabilirsiniz Şekil 2. Ssh ile bağlanmak için baxter ile aynı ağda olduğunuza emin olun. Labda bulunan Şekil 2 deki girişe sahip mavi ethernet kabloları ile cmpe ağına katılabilirsiniz. Herhangi bir kod çalıştırmadan önce baxter.sh'ı sourcelamak ve enable_robot kodunu çalıştırmak gereklidir. Enable_robot kodunu çalıştırılmayı unutursanız kod çalışıyor gözükse bile robot hareket etmeyecektir.

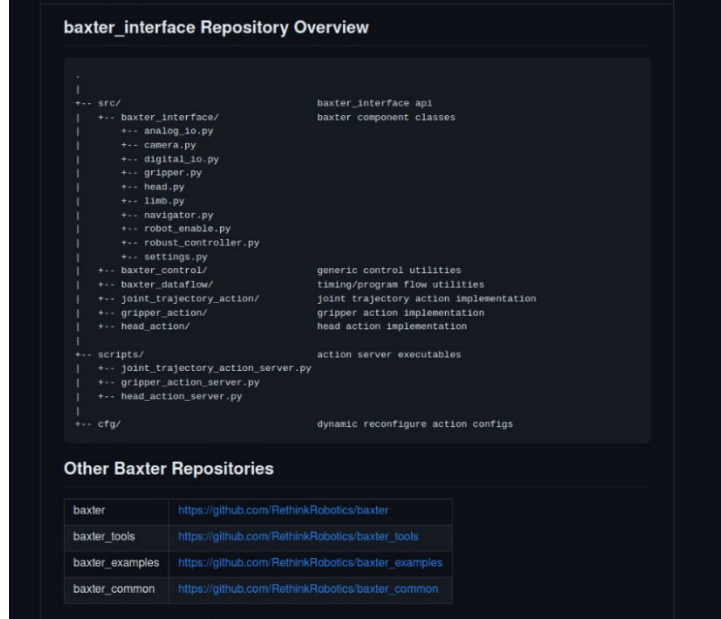
```
ruser@011607P0027:~  
colors@colors-MS-7E07:~$ ssh ruser@79.123.176.47  
Password:  
ruser@011607P0027 ~ $ cd ros_ws/  
ruser@011607P0027 ~/ros_ws $ source baxter.sh  
[baxter - http://localhost:11311] ruser@011607P0027 ~/ros_ws $ cd ..  
[baxter - http://localhost:11311] ruser@011607P0027 ~ $ rosrn baxter_tools enab  
le_robot.py -e  
[INFO] [WallTime: 1692111016.753630] Robot Enabled  
[baxter - http://localhost:11311] ruser@011607P0027 ~ $
```



Şekil 2

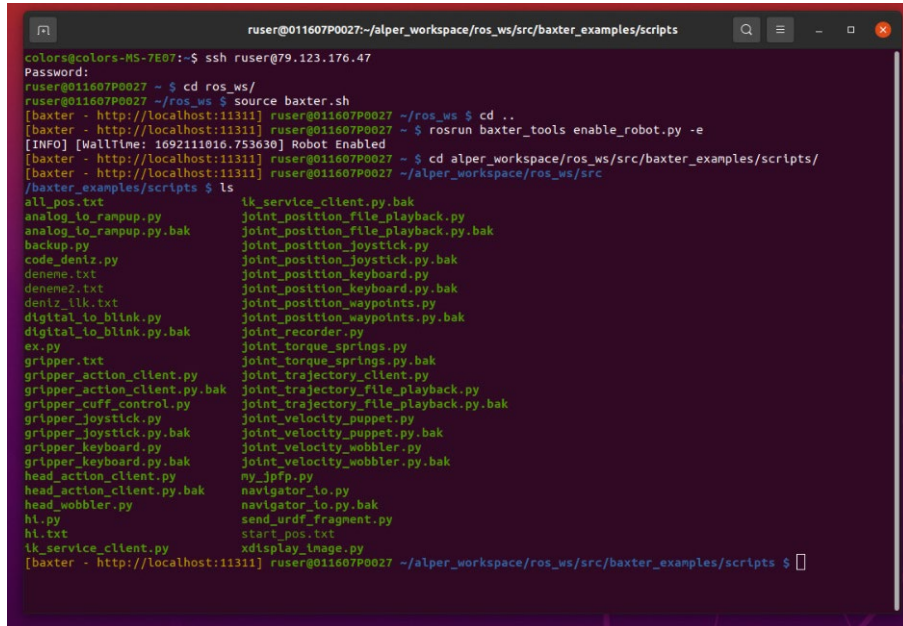
2)Kod çalıştırmak:

a) Baxter'in bilgisayarında kullanıcılara kolaylık olsun diye yazılmış baxter_interface veya baxter_examples (https://github.com/RethinkRobotics/baxter_interface) gibi çeşitli kodlar bulunmaktadır.



Şekil 3

Baxter'in bilgisayarında bulunan örnek kodları çalıştırmak için baxter_examples dosyasına gitmelisiniz. Tam konumu için Şekil 4. Burada ./dosya_adi.py ya da python dosya_adi.py komutları ile istediğiniz kodu çalıştırabilirsiniz. Örnek olarak "roslaunch baxter_examples head_wobbler.py" yazarak robotun kafasını sağa sola hareket etmesini sağlayabilirsiniz. Dikkat edilmesi gereken bir sorun bu kodlar python2.7 sürümünde yazılmıştır ve baxter'in bilgisayarındaki python da python2.7'dir.



Şekil 4

Aynı ağdaki başka bir bilgisayardan kod çalıştırmak için ise kendi bilgisayarınızın ROS_MASTER'ını baxter yapmalısınız. Bunu bilgisayarın ~/.bashrc'sine ros_master ve ros_ip'yi ekleyerek otomatize edebilirsiniz Şekil 5. Şekil 5'teki satırları eklemek için "nano ~/.bashrc" komutunu kullanıp açılan text editordan yazabilirsiniz. ROS_MASTER_URI "http://" + "baxter_ip" + ":11311" dir. Baxter'ın ip'si statik olduğundan bir kez ekledikten sonra değiştirmenize gerek yoktur. ROS_IP bilgisayarınızın cmpe ağındaki ip'si olmalıdır ama bu ip bilgisayarı kapatıp açtığınızda değişebileceği için her açılışta kontrol etmekte yarar vardır. Bilgisayarınızın ip'sini "ifconfig" komutu ile terminalden öğrenebilirsiniz. Cmpe ağında olan bilgisayarların ip'si 79.123.176.XX şeklindedir. **~/bashrc ye yaptığınız değişikliklerden sonra terminali yeniden başlatmayı unutmayın.** Eğer rostopic list komutunu çalıştırdığınızda topicleri görebiliyorsanız ama rostopic echo/topic_name yapınca topic'i göremiyorsanız problem bilgisayarınızın ip'sini güncellememekten kaynaklanıyor olabilir. Daha fazla bilgi için <http://wiki.ros.org/ROS/Tutorials/MultipleMachines> .

```
export ROS_IP=79.123.176.164

source /opt/ros/noetic/setup.bash
export ROS_MASTER_URI=http://79.123.176.47:11311
source ~/Desktop/catkin_ws/devel/setup.bash
```

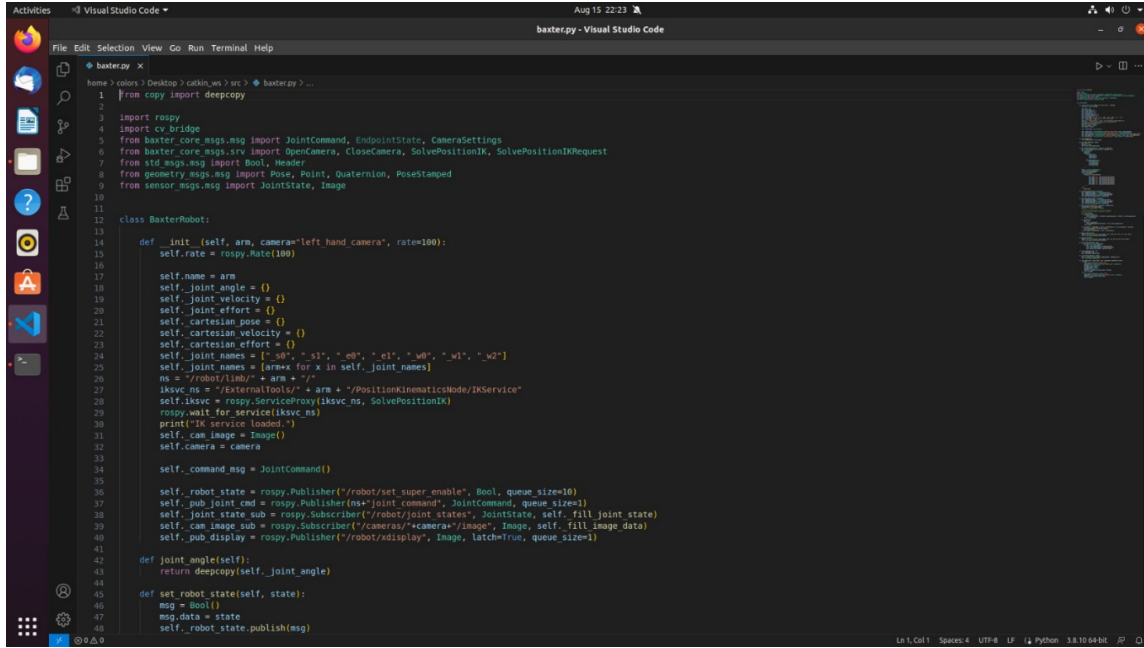
Şekil 5

Bu işlemi yaptıktan sonra rostopic list ile baxter'ın topiclerini görebiliyor olmanız lazım.

```
colors@colors-MS-7E07: ~$ nano ~/.bashrc
colors@colors-MS-7E07: ~$ rostopic list
/ExternalTools/left/PositionKinematicsNode/FKServer/reply
/ExternalTools/left/PositionKinematicsNode/FKServer/request
/ExternalTools/left/PositionKinematicsNode/IKServer/reply
/ExternalTools/left/PositionKinematicsNode/IKServer/request
/ExternalTools/right/PositionKinematicsNode/FKServer/reply
/ExternalTools/right/PositionKinematicsNode/FKServer/request
/ExternalTools/right/PositionKinematicsNode/IKServer/reply
/ExternalTools/right/PositionKinematicsNode/IKServer/request
/cameras/left_hand_camera/camera_info
/cameras/left_hand_camera/camera_info_std
/cameras/left_hand_camera/image
/cameras/right_hand_camera/camera_info
/cameras/right_hand_camera/camera_info_std
/cameras/right_hand_camera/image
/collision/left/collision_detection
/collision/left/debug
/collision/right/collision_detection
/collision/right/debug
/diagnostics
/diagnostics_agg
/diagnostics_toplevel_state
/hdraw
/robot/accelerometer/left/accelerometer/state
/robot/accelerometer/right/accelerometer/state
/robot/accelerometer_names
/robot/accelerometer_states
/robot/analog_io/command
/robot/analog_io/left_hand_range/state
/robot/analog_io/left_hand_range/value_uint32
/robot/analog_io/left_vacuum_sensor_analog/state
/robot/analog_io/left_vacuum_sensor_analog/value_uint32
/robot/analog_io/left_wheel/state
/robot/analog_io/left_wheel/value_uint32
/robot/analog_io/right_hand_range/state
/robot/analog_io/right_hand_range/value_uint32
/robot/analog_io/right_vacuum_sensor_analog/state
/robot/analog_io/right_vacuum_sensor_analog/value_uint32
/robot/analog_io/right_wheel/state
/robot/analog_io/right_wheel/value_uint32
/robot/analog_io/torso_fan/state
/robot/analog_io/torso_fan/value_uint32
/robot/analog_io/torso_left_wheel/state
/robot/analog_io/torso_left_wheel/value_uint32
/robot/analog_io/torso_lighting/state
/robot/analog_io/torso_lighting/value_uint32
/robot/analog_io/torso_right_wheel/state
/robot/analog_io/torso_right_wheel/value_uint32
/robot/analog_io_names
/robot/analog_io_states
/robot/assembly/head/state
/robot/assembly/left/state
```

Şekil 6

b) Kendi bilgisayarınızda python3 ile baxter_interface'i kullanamazsınız ama burdaki çoğu methodu baxter.py kodunda bulabilirsiniz. (Eğer topiclere kendiniz data göndermek istiyorsanız baxter_interface kullanmanıza gerek yok. Interface classlarla işi bir tık kolaylaştırıyor sadece.)



```
1 from copy import deepcopy
2
3 import rospy
4 import cv_bridge
5 from baxter_core_msgs.msg import JointCommand, EndpointState, CameraSettings
6 from baxter_core_srvs.srv import OpenCamera, CloseCamera, SolvePositionIK, SolvePositionIKRequest
7 from std_msgs.msg import Bool, Header
8 from geometry_msgs.msg import Pose, Point, Quaternion, PoseStamped
9 from sensor_msgs.msg import JointState, Image
10
11
12 class BaxterRobot:
13
14     def __init__(self, arm, camera="left hand camera", rate=100):
15         self.rate = rospy.Rate(100)
16
17         self.name = arm
18         self._joint_angle = {}
19         self._joint_velocity = {}
20         self._joint_effort = {}
21         self._cartesian_pose = {}
22         self._cartesian_velocity = {}
23         self._cartesian_effort = {}
24         self._joint_names = ["s0", "s1", "e0", "e1", "w0", "w1", "w2"]
25         self._joint_names = [name for x in self._joint_names
26                             if name != "/robot/lib/" + arm + "/"]
27         self.iksvc_ns = "/ExternalTools/" + arm + "/PositionKinematicsMode/IKService"
28         self.iksvc = rospy.ServiceProxy(iksvc_ns, SolvePositionIK)
29         rospy.wait_for_service(iksvc_ns)
30         print("IK Service loaded.")
31         self.cam_image = Image()
32         self.camera = camera
33
34         self._command_msg = JointCommand()
35
36         self._robot_state = rospy.Publisher("/robot/set_super_enable", Bool, queue_size=10)
37         self._pub_joint_cmd = rospy.Publisher("/robot/joint_command", JointCommand, queue_size=1)
38         self._joint_state_sub = rospy.Subscriber("/robot/joint_states", JointState, self._fill_joint_state)
39         self._cam_image_sub = rospy.Subscriber("/camera/" + camera + "/image", Image, self._fill_image_data)
40         self._pub_display = rospy.Publisher("/robot/display", Image, latch=True, queue_size=1)
41
42     def _joint_angle(self):
43         return deepcopy(self._joint_angle)
44
45     def _set_robot_state(self, state):
46         msg = Bool()
47         msg.data = state
48         self._robot_state.publish(msg)
```

Şekil 7

Baxter.py kodunu kullanmak için ise ros_workspace'inize (ros_ws nasıl oluşturulur: http://wiki.ros.org/catkin/Tutorials/create_a_workspace) ekteki linkten bulabileceğiniz https://github.com/RethinkRobotics/baxter_common baxter_core_msgs'i ekleyip catkin_make yapmalısınız. Bunun için repoyu clone'layıp sadece catkin_core_messages folder'ını src'nin altına atabilirsiniz. "catkin_make" komutunu "catkin_ws" folderının içinde çalıştırın. Bu reponun compiled olduğunu Şekil 8'den anlayabilirsiniz.


```
colors@colors-MS-7E07: ~/Desktop/catkin_ws
colors@colors-MS-7E07:~/Desktop/catkin_ws/src$ code .
colors@colors-MS-7E07:~/Desktop/catkin_ws/src$ cd ..
colors@colors-MS-7E07:~/Desktop/catkin_ws$ catkin_make
Base path: /home/colors/Desktop/catkin_ws
Source space: /home/colors/Desktop/catkin_ws/src
Build space: /home/colors/Desktop/catkin_ws/build
Devel space: /home/colors/Desktop/catkin_ws/devel
Install space: /home/colors/Desktop/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/colors/Desktop/catkin_ws/build"
####
#### Running command: "make -j32 -l32" in "/home/colors/Desktop/catkin_ws/build"
####
[ 0%] Built target std_msgs_generate_messages_py
[ 0%] Built target geometry_msgs_generate_messages_py
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_OpenCamera
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AssemblyStates
[ 0%] Built target sensor_msgs_generate_messages_py
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_JointCommand
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndEffectorCommand
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndpointStates
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CameraSettings
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_HeadPanCommand
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_DigitalOutputCommand
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CollisionAvoidanceState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CloseCamera
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndEffectorProperties
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_RobustControllerStatus
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CollisionDetectionState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndEffectorState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AnalogIOState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CameraControl
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_URDFConfiguration
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_DigitalIOStates
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AnalogIOStates
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_SEAJointState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AssemblyState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_DigitalIOState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_HeadState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_ListCameras
[ 0%] Built target sensor_msgs_generate_messages_nodejs
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndpointState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_NavigatorState
[ 0%] Built target std_msgs_generate_messages_cpp
```

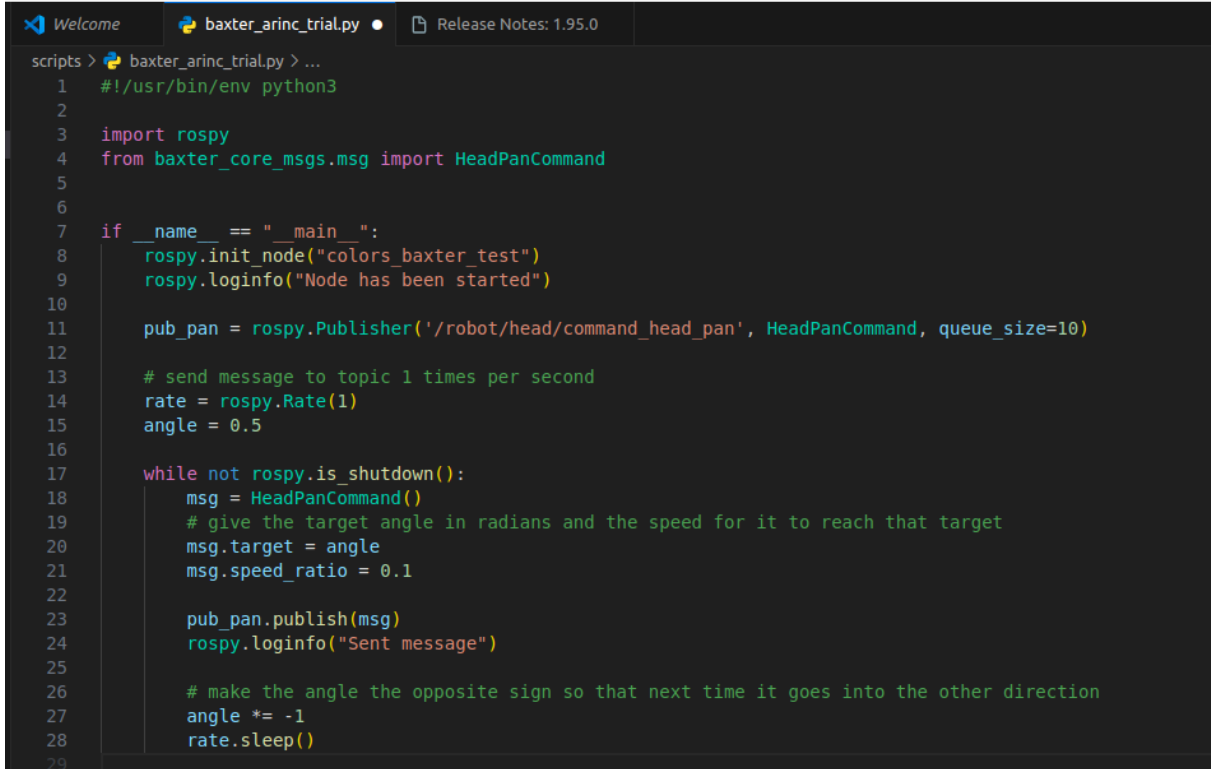
Şekil 8

Bu işlemden sonra devel dosyasındaki setup.bash'i sourelayarak kullanabilirsiniz.

```
colors@colors-MS-7E07: ~/Desktop/catkin_ws/devel
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CameraSettings
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_HeadPanCommand
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_DigitalOutputCommand
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CollisionAvoidanceState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CloseCamera
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndEffectorProperties
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_RobustControllerStatus
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CollisionDetectionState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndEffectorState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AnalogIOState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_CameraControl
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_URDFConfiguration
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_DigitalIOStates
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AnalogIOStates
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_SEAJointState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AssemblyState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_DigitalIOState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_HeadState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_ListCameras
[ 0%] Built target sensor_msgs_generate_messages_nodejs
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_EndpointState
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_NavigatorState
[ 0%] Built target std_msgs_generate_messages_cpp
[ 0%] Built target std_msgs_generate_messages_nodejs
[ 0%] Built target geometry_msgs_generate_messages_nodejs
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_AnalogOutputCommand
[ 0%] Built target sensor_msgs_generate_messages_cpp
[ 0%] Built target geometry_msgs_generate_messages_cpp
[ 0%] Built target std_msgs_generate_messages_eus
[ 0%] Built target sensor_msgs_generate_messages_lisp
[ 0%] Built target geometry_msgs_generate_messages_lisp
[ 0%] Built target std_msgs_generate_messages_lisp
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_SolvePositionIK
[ 0%] Built target geometry_msgs_generate_messages_eus
[ 0%] Built target sensor_msgs_generate_messages_eus
[ 0%] Built target baxter_core_msgs_generate_messages_check_deps_NavigatorStates
[ 38%] Built target baxter_core_msgs_generate_messages_cpp
[ 78%] Built target baxter_core_msgs_generate_messages_eus
[ 78%] Built target baxter_core_msgs_generate_messages_py
[100%] Built target baxter_core_msgs_generate_messages_lisp
[100%] Built target baxter_core_msgs_generate_messages_nodejs
[100%] Built target baxter_core_msgs_generate_messages
colors@colors-MS-7E07:~/Desktop/catkin_ws$ cd devel/
colors@colors-MS-7E07:~/Desktop/catkin_ws/devel$ source setup.bash
colors@colors-MS-7E07:~/Desktop/catkin_ws/devel$
```

Şekil 9

Şimdi interface kullanmadan komut göndermeye örnek yapalım. Şekil 10 daki kodu yarattığınız bir package'in içinde bir python dosyasına atarsanız ve çalıştırırsanız baxter kafasını sağa sola sallayacak. Çalıştırmak için önce bir "catkin_make" yapın sonra da "roslaunch your_package_name your_file_name" yapabilirsiniz. Tebrikler, Baxter'i kodla kontrol ettiniz! Gönderebileceğiniz diğer komutlar için "rostopic list" ile topic seçip "rostopic info topic/name" ile message type'ını bulabilirsiniz. "rostopic show message/name" ile message type'ın içeriğine bakabilirsiniz.

The image shows a code editor window with a dark theme. The title bar at the top has three tabs: 'Welcome', 'baxter_arinc_trial.py', and 'Release Notes: 1.95.0'. The main editor area displays a Python script. The script starts with a shebang line, imports 'rospy' and 'HeadPanCommand' from 'baxter_core_msgs.msgs'. It then checks if it's the main module, initializes a node named 'colors_baxter_test', and publishes to the '/robot/head/command_head_pan' topic. A loop publishes messages at 1 Hz, alternating the target angle between 0.5 and -0.5 radians. The script is as follows:

```
1  #!/usr/bin/env python3
2
3  import rospy
4  from baxter_core_msgs.msgs import HeadPanCommand
5
6
7  if __name__ == "__main__":
8      rospy.init_node("colors_baxter_test")
9      rospy.loginfo("Node has been started")
10
11     pub_pan = rospy.Publisher('/robot/head/command_head_pan', HeadPanCommand, queue_size=10)
12
13     # send message to topic 1 times per second
14     rate = rospy.Rate(1)
15     angle = 0.5
16
17     while not rospy.is_shutdown():
18         msg = HeadPanCommand()
19         # give the target angle in radians and the speed for it to reach that target
20         msg.target = angle
21         msg.speed_ratio = 0.1
22
23         pub_pan.publish(msg)
24         rospy.loginfo("Sent message")
25
26         # make the angle the opposite sign so that next time it goes into the other direction
27         angle *= -1
28         rate.sleep()
29
```

Şekil 10