Intel® Euclid™ Development Kit
Operating Guide

Intel Corporation
2200 Mission College Blvd
Santa Clara, CA USA 95054
www.intel.com
# Table of Contents

1.0 Box Contents ......................................................................................... 4
1.1 Product Introduction ........................................................................ 5
1.2 Attaching the Battery ....................................................................... 5
1.3 Powering ON the device ................................................................. 6
1.4 Powering OFF the device ................................................................. 6
1.5 Quick Start Instruction ..................................................................... 6
1.6 Web Interface .............................................................. ............................. 7
  1.6.1 Scenarios Tab .............................................................................. 7
  1.6.2 Euclid Nodes Tab ........................................................................ 9
  1.6.3 Monitor Tab ................................................................................ 11
  1.6.4 Configuration Tab ............................................................... 12
  1.6.5 Utilities Tab .............................................................................. 13
  1.6.6 About ...................................................................................... 17
  1.6.7 Power ...................................................................................... 17
1.7 Remote Connection ........................................................................... 17
  1.7.1 VNC Setup ............................................................................... 17
  1.7.2 SSH Setup ............................................................................... 17
  1.7.3 Local Setup ............................................................................... 17
1.8 Working with a ROS Companion Machine ....................................... 18
  1.8.1 DEVICE as a ROS Master ..................................................... 18
  1.8.2 DEVICE as a ROS device .................................................... 18
1.9 Working with and Arduino Device .................................................. 19
1.10 Adding a new ROS package .......................................................... 20
2.0 Sample Euclid Nodes ..................................................................... 20
  2.1 Cameras ...................................................................................... 20
  2.2 Cameras No Motion ...................................................................... 21
  2.3 RealSense SLAM ......................................................................... 21
  2.4 Depth Transcode .......................................................................... 21
  2.5 Depth Follower ............................................................................ 21
## Table of Contents (cont.)

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6</td>
<td>Collision Avoidance</td>
<td>21</td>
</tr>
<tr>
<td>2.7</td>
<td>Person Follower</td>
<td>22</td>
</tr>
<tr>
<td>2.8</td>
<td>Robot Movement Controller</td>
<td>22</td>
</tr>
<tr>
<td>2.9</td>
<td>RealSense Person Tracking</td>
<td>22</td>
</tr>
<tr>
<td>2.10</td>
<td>Static Transform</td>
<td>22</td>
</tr>
<tr>
<td>2.11</td>
<td>Arduino</td>
<td>23</td>
</tr>
<tr>
<td>2.12</td>
<td>Turtlebot</td>
<td>23</td>
</tr>
<tr>
<td>3.0</td>
<td>Sample Scenarios</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>6 Degrees of Freedom (6DoF)</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>6DoF-Person</td>
<td>24</td>
</tr>
<tr>
<td>3.3</td>
<td>Arduino CA—Collision Avoidance</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Arduino Person Follower</td>
<td>24</td>
</tr>
<tr>
<td>3.5</td>
<td>Cameras</td>
<td>25</td>
</tr>
<tr>
<td>3.6</td>
<td>Person View</td>
<td>25</td>
</tr>
<tr>
<td>3.7</td>
<td>Turtlebot CA—Collision Avoidance</td>
<td>25</td>
</tr>
<tr>
<td>3.8</td>
<td>Turtlebot Depth Follower</td>
<td>25</td>
</tr>
<tr>
<td>3.9</td>
<td>Turtlebot Person Follower</td>
<td>26</td>
</tr>
<tr>
<td>4.0</td>
<td>Turtlebot Follower Tutorial</td>
<td>26</td>
</tr>
</tbody>
</table>
1.0 Box Contents

The Intel® Euclid™ Development Kit (also known as Intel® Euclid™ Developer Kit) contains everything that you need to begin using the device right out of the box:

- Intel™ Euclid® Development Kit main unit (Main Unit)
- Lithium Polymer battery
- Power Adapter Plate
- USB 3.0 Cable (Type A to Micro B Super Speed)
- USB Power Cable (Type A to mini plug)
- AC Power Adapter (USB output with converters)
- HDMI Cable (Type D to Type A)
1.1 Product Introduction

1.2 Attaching the Battery

To attach the battery, align the pins on the battery to the contacts on the Main Unit. Slide the retaining hook into the slot next to the contacts, then gently push the battery onto the Main Unit until the retaining latch engages.
1.3 Power ON the Device

The Intel® Euclid™ Development Kit contains three methods of powering on the Main Unit.

1. Battery power. To use the device in its mobile mode, attach the battery as shown. Follow the quick start instructions.

2. Battery charging. To charge the battery, the device must be ON. Attach the battery to the Main Unit. Plug the AC Power Adapter into an available outlet. Connect the USB 3.0 cable into the AC Power Adapter with the other end into the OTG port on the Main Unit. Follow the quick start instructions.

3. No battery. To use the device without the battery connected, plug the AC Power Adapter into an available outlet. Attach the Power Adapter Plate to the back of the Main Unit (same procedure as attaching the battery). Connect the USB Power Cable to the AC Power Adapter and the Power Adapter Plate. The device is ready to power on.

**WARNING:** DO NOT attempt to supply power to the USB 3.0 Type A port. Only use the supplied Power Adapter and cable.

1.4 Power OFF the device

To power off the device, there are three methods:

1. Web interface. From the web interface, click on POWER, then choose SHUTDOWN.

2. Shutdown. Press and hold the ON/OFF button for ~8-10 seconds and release. The device will shut down.

3. Forced Shutdown. Press and hold the ON/OFF button for ~15 seconds. The device will force shut-down.

1.5 Quick Start Instructions

The Intel® Euclid™ Development Kit provides a “zero-installation” usage as all of the software needed to operate the device is pre-installed. For first time usage, please follow these instructions:

1. Attach the battery to the Main Unit. It is advised to connect the battery charger until the battery status is known.

2. Power-on the device by pressing the POWER ON/OFF BUTTON for 3 seconds. The SYSTEM ON LED will illuminate green. The CHARGING LED will illuminate red if the battery is being charged.

3. Wait for approximately 30 seconds until you hear the chime tone indicating that the device is ready to use.

4. Power-on your personal device—either a laptop, tablet, smart phone or other Wi-Fi enabled device. Please be sure that your personal device is within 3 meters of the Intel® Euclid™ Development Kit.

5. Access the settings of your personal device and use the wireless setup facility to connect your personal device to the Intel® Euclid™ Development Kit. Find the SSID “EUCLID_XXXX” and connect to it *(where XXXX is an alphanumeric code that is unique to your individual device)*. The password
that you will need is “12345678”.

6. Open the browser on your personal device and enter the following IP address: “http://10.42.0.1/” and hit ENTER. This will open the main portal for the Intel® Euclid™ Development Kit.
   
   **Note:** On some devices, you will get a prompt that there is not internet on this network, do you still want to connect. Please select yes and you can access Euclid’s web interface.

7. By using the device, you are accepting the software license. A complete version of the license can be viewed at: http://click.intel.com. After reading the license terms, please click to accept the software license.

8. **NOTE:** The root (admin) user for the Ubuntu machine is “euclid” and the password is “euclid” as well.

### 1.6 The Web interface - Overview

After logging in to the main portal, you will see the Intel® Euclid™ Web UI where the user can monitor the device status, create, edit and launch different scenarios and change configurations. This Web UI is a portal to control the Intel® Euclid™ Development Kit Main Unit (referred to as DEVICE), as well as control any robot connected to the DEVICE. Important device information such as the battery status and charging status is also found on the home page. Each tab of the UI will be reviewed in subsequent sections of this document.

![Web Interface](image.png)

A node is an executable file within a ROS package. ROS nodes are executables that use a ROS client library to communicate with other ROS nodes. ROS Nodes can publish or subscribe to a Topic. ROS nodes can also use or provide Services. In a robot, ROS nodes can be used to control various functions of the robot. ROS nodes provide fault tolerance as crashes are isolated to individual nodes and reduce code complexity. A ROS node is written with the use of a ROS client library, such as roscpp or rospy.

A Euclid Node is an abstraction for the Euclid automation layer in order to provide high level functionalities. It is in the essence a ROS launch file (which contains ROS nodes) and configuration arguments. Euclid Nodes can be created by the web interface and ROS nodes can be added to them. Sample Euclid Nodes have been provided, but users can easily define their own Euclid Nodes.

### 1.6.1 Scenarios Tab

The Scenario tab allows the user to configure scenarios, create new scenarios, start/stop scenarios and remove existing scenarios. A Scenario is a powerful way to control the DEVICE. A Scenario is an experience or application that can run on the DEVICE and is a set of Euclid Nodes that run together.

To define a Scenario more precisely, a Scenario is defined by a set of Euclid Nodes running together their respective configurations, the pose of the DEVICE with respect to the robot, and the underlying communication (native ROS or Arduino based). For example, a scenario can be defined as a ROBOT FOLLOWER by selecting to run the camera node, the follower node, and the robot’s ROS node.

Only one scenario can run at any one time. A scenario can be marked to ‘run on startup’ to cause the system to load that specific scenario as the system starts.

**Running Scenarios**—To choose a pre-defined scenario, click on that scenario from the Scenarios Tab, then click the Play button. The device will play a short tune to signal that the scenario
is running. A running scenario is shown by the scenario name being highlighted in green. To learn more about the scenario, there is a menu icon next to the scenario name. The NODES option will show the nodes running under the scenario. When a scenario fails to run, the scenario title box turns red. Clicking play on a different scenario while a first scenario is running stops the first scenario and starts the second one.

**Stopping scenarios**—To stop a running scenario, simply click on the STOP button below the scenario name.

**Configuring Scenarios**—Once a scenario or nodes are running, they can be configured via the Configuration Tab. The communication with the robot can be selected (ROS native –turtlebot – or — general serial Arduino connection). The pose of the DEVICE with respect to the robot can also be configured.

**Creating Scenarios**—To create a scenario, click on the CREATE NEW SCENARIO button, then enter a scenario name and select the desired Euclid Nodes, adding them one at a time. Sample nodes are provided or new nodes can be created. All of the selected Euclid Nodes will run when the scenario is launched. To configure any of these nodes, use the Configuration tab. For example, to create a Turtlebot follower scenario, select the camera node and choose the native Turtlebot robot. Once created, the new scenario is show on the Scenarios tab. The launch parameters for the nodes can also be specified while creating the scenario.
Remove a scenario—In order to remove an existing scenario, simply click on the menu icon on the scenario to be removed and click Delete.

Edit a scenario—In order to edit an existing scenario, simply click on the menu icon on the scenario to be removed and click Edit. You can change the name, nodes and launch parameters for the scenario.

Setting a default Scenario—To set a scenario as default, and to have the scenario launch on system startup, click on the star button. The button will turn to yellow for a default scenario. Clicking it again will remove the scenario from startup. Note: Only one scenario can be selected as the default scenario.

1.6.2 Euclid Nodes Tab

A Euclid Node differs from a ROS node. A Euclid Node is a launch file that can run a single or multiple ROS nodes. The main difference between the two is that a Euclid Node can contain specific launch parameters and for each of the underlying ROS nodes. The user can dynamically reconfigure the Euclid Nodes and save the changes within a scenario. A Euclid Node can be considered a module for specific features. For example, is there is a Euclid Node titled Cameras that will run the cameras.

On the Euclid Nodes page, you can add, edit, or delete Euclid Nodes from the ROS package(s) that have been installed on the device. To include user-defined ROS packages on the device and within the web interface, the user must load the ROS package(s) and any launch files within the launch folder.

Adding a new Euclid Node—To add a new Euclid Node, start on the Euclid Nodes tab and click on the
REGISTER NEW NODE button. The new Euclid Node can use the pre-existing or user-loaded ROS packages. Enter a name for the Euclid Node. The name can leverage the ROS node name, or it can be completely different. Then enter the name of the ROS package where the ROS node resides.

After clicking NEXT, a list of launch files from the specified package will be presented. Pick the desired launch file from the ROS package. The ROS nodes will be specified in the launch file.

After selecting the desired launch file and clicking NEXT, select the running ROS nodes associated with the Euclid Node that is being added. The selected ROS nodes will be monitored when running the Euclid Node. The system will run ROS nodes even if they are unselected (selecting is for monitoring
purposes). Launch parameters can be specified for the running ROS nodes by clicking on the Parameter buttons and ADDing them. Click SAVE and the node will be added the Web UI and be available for use in creating scenarios.

**Editing a Euclid Node**—To edit a Euclid Node, click on the node to be edited from the Euclid Nodes Tab. Then click on the menu icon to the right of the node name and select Edit and follow the wizard as for creating a new node (see above).

![Register Euclid Node](image)

**Deleting a Euclid Node**—To delete a Euclid Node, click on the node to be deleted from the Euclid Nodes Tab. Then click on the menu icon to the right of the node name and select Delete.

### 1.6.3 Monitor Tab

On the Monitor tab, the user can view previews from a running scenario. Only the feeds that are part of the active scenario will be visible. At the top of the page, you will see toggle buttons to turn on the various feeds: color (rgb camera), depth, fisheye, person (tracking), imu, and trajectory(6 DoF).

The Configurable Nodes button provides a drop down menu with the configurable items for the active nodes. Changing a configurable item will dynamically reconfigure the node—which uses the ROS dynamic reconfiguration feature.
1.6.4 Configuration Tab

From the Configuration Tab, each running node in the system can be independently configured. For example, the camera node can be configured with a white balance value, the laser emitter can be turned ON or OFF, or you can change the saturation on the color feed. To change a configuration, click on the CONFIGURABLE NODES box and choose the desired node from the dropdown list. (If no nodes are running, then the drop-down menu will be empty). The changeable parameters for that particular node will be visible. After the values of desired node parameter(s) are changed, click on UPDATE button. The changes are applied instantly. Clicking RESET will reset the parameters to default. Sometimes it may take several moments for the nodes to run as they are started sequentially. Click on the ‘refresh’ button to update the list. The TURN SLIDER OFF button will toggles the method of input from sliders to text box. After setting the configuration values, you can click on the SAVE CONFIGURATION button. The updated configuration values will load the next time the scenario is loaded. However, the modified configuration values are ONLY applied to the scenario in which the values were changed.
1.6.5 Utilities Tab
The utilities tab gives the user access to most of the Intel® Euclid™ Development Kit system settings.

System Status
The System Status tab provide information about the Wi-Fi status, the hardware information, the connected USB devices, and the ROS Master URI.
Nodes Status
The Nodes Status Tab provides basic monitoring functionality. All the nodes should be running, which will be indicated by a green status box next to the node name. If one of the nodes is not running, this may cause unexpected behavior. To remedy, first try re-launching the scenario, and if that fails, restart the device. If further debugging is needed, check the system log page to determine the root cause of the failure. Only nodes that are part of the active scenario will display under Scenario Nodes.
**Wifi Settings**

The Wifi Settings Tab allows the user to connect to a WLAN or to enable the hotspot.

To connect to an external Wifi network, click on the SCAN button. This will disconnect the device and scan for nearby Wifi devices. It may take up to 1 minute to complete. Re-establish the Wifi connection between your personal device and the DEVICE, then refresh the browser. The available Wifi networks will be shown under Available Networks. After connecting to the network, you may need to enter the password for the network.

If the device had previously connected to a network, they will be listed under the Saved Networks list.

Note that when the device connects to another network, it will be assigned an IP address by that network. The user will need to determine this new IP address to enter it into the browser to view the web UI. One way to determine the IP address is to connect the device to a display and keyboard—and then running “`ifconfig`” in a terminal.

In order to be identified on network, the device can publish its IP address using a standard multicast system. The following is an example that shows how to consume the multicast messages on a smartphone.

1. On the phone install a multicast app such as ‘Multicast Tester’.
2. Connect your phone and the Euclid device to a single network.
3. Open the multicast application on your phone and enter the following
   - Multicast IP: 224.1.1.100
   - Multicast Port: 10000
   - Start Listening to multicast messages.
   - Messages will appear with the corresponding device hostname and its IP address.

**Note:** The multicast app will show every Intel Euclid Development Kit that is connected to the current network.
**Logs**

On the Logs Tab, the user can see the ROS logs, any errors and information for the nodes that are running.

**Arduino Lib**

The Arduino Lib Tab contains an interface to generate or download an Arduino library for the device. The GENERATE NEW LIBRARY button will start the library generation. After the library generation is complete, click on the DOWNLOAD LIBRARY button to download a zip file of the library. The file should be extracted to the Arduino IDE Libraries directory.
1.6.6 About Tab
The About tab provides two sub-tabs. The first is the License sub-tab that provides license information on the various software that is pre-installed on the device. Help is the second sub-tab and it will show an overview of the Intel Euclid Development Kit.

1.6.7 Power Tab
The Power tab provides three options. The first is Restart OOBEC, which will reset the Euclid device to it’s out-of-the box state. The second is Reboot, which will restart the device. The third is the Shutdown, which will turn off the device. Lastly, the battery charge state is shown at the bottom of the menu.

1.7 Remote Connection
The Intel Euclid Development Kit can also be used in a remote connection mode. To begin using the device in a remote mode, turn on the device as noted in sections 1.3 and 1.5 of this manual.

1.7.1 VNC Setup
Virtual Network Computing (VNC) is a graphical desktop sharing system that uses the Remote Frame Buffer protocol to remotely control another computer. A VNC server transmits a duplicate of the remote computer’s display to the VNC client. The VNC server receives, interprets and executes commands out from the VNC client on the remote computer.

The Intel Euclid Development Kit operates as a VNC server out of the box. To operate in this mode, launch the VNC client on the personal device (i.e. laptop, tablet, smartphone). Enter the device IP (10.42.0.1) and connect. The password to complete the connection is “euclid”.

1.7.2 SSH Setup
Since Euclid runs Ubuntu, SSH (Secure Shell) can be used natively. To run SSH, first install SSH on the Intel Euclid Development Kit. Second, on the second Linux machine, type in the following command:

```
ssh <Intel Euclid Development Kit IP address>
```

1.7.3 Local Setup
The Intel Euclid Development Kit was designed to be operated in a local setup mode. To setup the device in this way, connect a USB hub to the USB 3.0 port and connect a keyboard and mouse to the hub. Connect an external monitor to the device via the HDMI port and the supplied HDMI cable. Finally, connect the device to one of the power supply methods (battery power, or connect the USB OTG port to the included power supply, or connect the Power Adapter Plate and power supply).
1.8 Working with a ROS Companion Machine

There are two options working with the DEVICE:

1.8.1 DEVICE as a ROS Master

Euclid is running ROSCore by default. To configure your client device to connect to the DEVICE’s Hotspot, follow those steps:

ROSCORE is a collection of nodes and programs that are pre-requisites of a ROS-based system. You must have a roscore running in order for ROS nodes to communicate. It is launched using the roscore command.

1. Configure the ROS_MASTER_URI to 10.42.0.1 (the DEVICE’s hotspot IP)
$> \text{export } \text{ROS\_MASTER\_URI}=10.42.0.1$

2. Run "ifconfig" to get the IP address and set it to ROS_IP:
$> \text{export } \text{ROS\_IP}=<\text{Machine IP}>

1.8.2 DEVICE as a ROS device

The DEVICE can be configured to connect to an existing ROS Master device. First configure the DEVICE to connect to the desired wireless network (see section 1.6.5 Wifi Settings), then follow these steps:

1. Connect to the web interface
2. Navigate to the system settings page
3. Change the ROS MASTER URI to the newly assigned IP address
1.9 Working with and Arduino Device

The DEVICE comes with a native support for Arduino, based on `ros_bridge`, `ros_serial` and `ros_arduino`.

The DEVICE sends data to a companion Arduino device via serial interface (USB or Bluetooth). There are many examples how to use ROS and Euclid directly from Arduino. The examples are shown in the screenshot on page 20, and include two created specifically for the DEVICE (denoted “Euclid___”). In order to use the DEVICE with an Arduino machine:

1. On a device with Arduino IDE installed, connect to the Web interface (see section 1.5)
2. Navigate to the Arduino page an click on “Generate a new library"
3. Download the new library. It will take a few seconds until the notification to start the download appears.
4. Open the IDE and click on the IMPORT .ZIP LIBRARY.
5. Select the downloaded ZIP library and follow the instructions on the IDE.
6. To get started, choose any of the provided samples:
1.10 Adding a new ROS package

You can add a new package to the DEVICE by either of the following:

- Install a package from an external source (apt-get for example)
- Compile the ROS package directly on the device

Once the ROS package is working on the DEVICE, add it to the web interface:

1. Go to the Euclid Nodes Tab, then click on Register New Node
2. Follow the instructions to add the new node.

2.0 Sample Euclid Nodes

The following will review the sample Euclid Nodes

2.1 Cameras

The Cameras Node is for getting data from all the Camera Sensors – RGB, Fisheye, and Depth and IMU. The camera parameters can be changed from the Configuration Tab, under the configurable nodes drop down, and selecting “/camera/driver”. The various camera parameters can be changed (i.e. emitter, auto exposure, saturation, brightness, etc.).

- **Package** – realsense_camera
- **Launch File** – lr200m_nodelet_default.launch

2.2 Cameras No Motion

The Cameras No Motion Node is used to get data from the Color and Depth Camera only. The fisheye
and IMU are disabled. The camera parameters can be changed from the Configuration Tab, under the configurable nodes drop down and selecting “/camera/driver”. The various camera parameters can be changed (i.e. emitter, auto exposure, saturation, brightness, etc.).

- **Package** – realsense_camera
- **Launch File** – lr200m_nodelet_default.launch
- **Launch Parameters:**
  - enable_fisheye:= false
  - enable_imu:= false

### 2.3 RealSense SLAM

The RealSense SLAM Node runs the SLAM algorithm using the fisheye and the IMU data and provides the pose data for 6DOF tracking.

- **Package** – realsense_sp
- **Launch File** – sp.launch

### 2.4 Depth Transcode

The Depth Transcode node converts the depth data from the depth camera to a viewable format. This node is essential if you want to see the Depth data on the Web Viewer.

- **Package** – cs_depth_transcode
- **Launch File** – depth_transcode_nodelet.launch

### 2.5 Depth Follower

The Depth Follower node is used to make a robot follow any closest depth based blob within the detectable range of the field of view of the camera. The node parameters can be changed in the Configuration Tab by selecting /realsense_depth_follower from the drop down menu. Parameters such as the size of the detected blob, and the size of the detection camera frustum (the minimum and maximum X, Y, Z range) can be changed. The node takes the depth data from the camera node and sends the x, y, z goal to the Robot movement Controller node.

- **Package** – cs_depth_transcode
- **Launch File** – depth_transcode_nodelet.launch

### 2.6 Collision Avoidance

The Collision Avoidance node runs an algorithm which makes the robot detect an obstacle using the depth data. If the DEVICE finds an obstacle in the detectable depth frustum, it causes the robot to make a random left or right turn. This nodes takes the depth data from the camera nodes and generates a movement goal (x, y, z) for the Robot Movement Controller node. The node parameters can be changed in the Configuration Tab by selecting realsense_wanderer from the drop down menu. The changeable items include the detection frustum and the distance the robot can get to the detected obstacle before turning.

- **Package** – realsense_wanderer
- **Launch File** – realsense_wanderer.launch

### 2.7 Person Follower
In the Person Follower node, the robot follows the first detected person in the field of view of the camera. This node takes the data from the RealSense Person Tracking node and sends a goal to the Robot Movement Controller node. This node in turn tells the robot to follow at a certain distance.

**Package** – realsense_person_follower  
**Launch File** – realsense_person_follower.launch

### 2.8 Robot Movement Controller

The Robot Movement Controller node receives a target goal (x, y, z position) from various other nodes (such as the Depth Follower, Person Follower, and Collision Avoidance nodes). The Robot Movement Controller node sends cmd_vel messages to the specific robot base. The follow distance and other parameters can be changed in the Configuration Tab by selecting “/realsense_robot_controller” from the Configurable Nodes dropdown.

**Package** – robot_movement_control  
**Launch File** – robot_movement_controller.launch

### 2.9 RealSense Person Tracking

The RealSense Person Tracking node detects people in the field of view of the camera and uses the depth and color data from the camera node.

**Package** – realsense_person  
**Launch File** – person_sample.launch

### 2.10 Static Transform

The DEVICE supports static transformation between two TF frames. Users can add transforms directly from the web.

**Package** – configuration_node  
**Launch File** – static_transform.launch  
**Launch Parameters** -  
- x:=1  
- y:=0  
- z:=0  
- Yaw:=0  
- Pitch:=0  
- Roll:=0  
- Frame_id:="camera_link"  
- Child_frame_id:= “base_link”  
- Frame_rate:-100

### 2.11 Arduino

The Arduino node enables a Arduino-based robot that communicates over a serial port (USB). The parameters of the Arduino-based robot can be changed within the Configuration Tab by selecting “/ CsArduinoMotorConfiguration” from the configuration item drop down. This node will receive cmd_vel messages from the Robot Movement Controller Node.

**Package** – cs_arduino_wrapper
Launch File – arduino_wrapper_nodelet.launch
Launch Parameters-
    port:=/dev/ttyACM0

2.12 Turtlebot

The Turtlebot node selects a Turtlebot 2 base to be run with other nodes. This node will receive cmd_vel messages from the Robot Movement Controller Node.

    Package – turtlebot_bringup

3.0 Sample Scenarios

The following are sample scenarios that are pre-installed on the DEVICE.

3.1 6 Degrees of Freedom (6DoF)

The 6DoF scenario demonstrates the SLAM and special tracking capabilities of the DEVICE. On the Scenario Tab, start the scenario and then go to the Monitor Tab and turn on the Fisheye and Trajectory views. For best performance, it is recommended that you start the scenario with the device stationary. This allows the fisheye camera to efficiently capture features and minimizes drift in the trajectory. The DEVICE can move to capture the trajectory in the viewer.

    Euclid Nodes
        1. Cameras – To get the camera data
        2. RealSense SLAM – To get the pose
        3. Depth Transcode – To make the depth data viewable

3.2 6DoF-Person

The 6DoF-Person scenario is same as the 6DoF scenario, with the only difference being the addition of the person tracking Euclid Node. On the Scenario Tab, start the scenario and then go to the Monitor Tab and turn on the Person, Fisheye and Trajectory views.

    Euclid Nodes
        1. Cameras
        2. RealSense SLAM
        3. Depth Transcode
        4. RealSense Person Tracking - to find the people in the field of view of camera

3.3 ArduinoCA – Collision Avoidance

The ArduinoCA is a scenario that uses a ROS+Arduino-based robot connected via USB. The robot will run randomly avoid obstacles. Once it sees an obstacle, it will turn left or right randomly. The communication with the Arduino happens over a serial port. The parameters in the can be adjusted in the Configuration Tab. For example, the detection frustum can be changed to adjust the field of view of the robot for the optimum performance and collision avoidance.

    Euclid Nodes
        1. Cameras
        2. Depth Transcode
        3. Collision Avoidance – Detects and avoids obstacles
4. Robot Movement Controller – Sends cmd velocity messages to the robot
5. Arduino

3.4 ArduinoPersonFollower

The ArduinoPersonFollower scenario is for using an Arduino+ROS based robot to follow the tracked
person. The robot will maintain a distance of 1.5 meters distance from the person being tracked.

Euclid Nodes
1. Cameras
2. Depth Transcode
3. Person Follower
4. Robot Movement Controller
5. Realsense Person Tracking
6. Arduino

3.5 Cameras

The Cameras Scenario runs all the cameras. To view the camera(s) stream, navigate to the Monitor
Tab and turn on the color, depth, fisheye, and IMU.

Euclid Nodes
1. Cameras
2. Depth Transcode

3.6 PersonView

The PersonView scenario runs the camera node along with the person tracking node to track people in
the field of view of the camera.

Euclid Nodes
1. Cameras
2. Depth Transcode
3. RealSense Person Tracking

3.7 TurtlebotCA – Collision Avoidance

The TurtlebotCA scenario provides collision avoidance using a Turtlebot 2. Plug the DEVICE in to a
Turtlebot 2 using the provided USB cable. Mount the Device to the Turtlebot 2 so that the DEVICE is
on the front of the top plate looking forward. Make sure that the view of the DEVICE is not obstructed.
Run the Scenario and the Turtlebot 2 will start moving forward, and when it sees an obstacle it will
turn left or right avoiding it.

Euclid Nodes
1. Cameras
2. Depth Transcode
3. Collision Avoidance
4. Robot Movement Controller
5. Turtlebot

3.8 TurtlebotDepthFollower

The TurtlebotDepthFollower Scenario detects a blob in the depth camera image in Euclid and follows
it maintaining a certain distance. Plug the DEVICE in to a Turtlebot 2 using the provided USB cable. Mount the Device to the Turtlebot 2 so that the DEVICE is on the front of the top plate looking forward. Make sure that the view of the DEVICE is not obstructed. The DEVICE should be connected to the Turtlebot 2 as shown in the image. The follow distance can be modified on the Configuration Tab.

**Euclid Nodes**
1. Cameras
2. Depth Transcode
3. Depth Follower
4. Robot Movement Controller
5. Turtlebot

### 3.9 TurtlebotPersonFollower

The TurtlebotPersonFollower scenario will cause a Turtlebot follow the user at 1.5 meters. The distance is configurable in the Configuration Tab by changing the parameter in the person follower node. Plug the DEVICE in to a Turtlebot 2 using the provided USB cable. Mount the Device to the Turtlebot 2 so that the DEVICE is on the front of the top plate looking forward. Make sure that the view of the DEVICE is not obstructed. After starting the scenario, the robot will follow the user.

**Euclid Nodes**
1. Cameras
2. Depth Transcode
3. Person Follower
4. Robot Movement Controller
5. RealSense Person Tracking
6. Turtlebot

### 4.0 Support Resources

For additional resources, click on the following link and navigate to the Intel® Euclid™ Development Kit: [http://click.intel.com](http://click.intel.com)

To get support for an Intel® Euclid™ Development Kit, navigate to the community support page: [https://communities.intel.com/community/tech/realsense](https://communities.intel.com/community/tech/realsense)

To join the Intel® Euclid™ Development Kit community to share and learn: [http://euclidcommunity.intel.com/](http://euclidcommunity.intel.com/)
Intel® Euclid™ Development Kit

featuring:

Manufactured by:
Intel Corporation
2200 Mission College Blvd.
Santa Clara, CA 95054   USA
Intel® Euclid™ Development Kit, Model CCS-G003, mm# 956912

Ubuntu® is a registered trademark of Canonical Ltd.
Wi-Fi® is a registered trademark of the Wi-Fi Alliance.
Bluetooth® is a registered trademark of Bluetooth SIG, Inc.
HDMI® is a registered trademark of HDMI Licensing Administrator, Inc.
USB® is registered trademark of USB Implementers Forum, Inc.