<https://answers.ros.org/question/371902/how-to-subscribe-to-laser-scan-topic-and-publish-a-customized-scan-topic-back/>

<https://gist.github.com/atotto/c47bc69a48ed38e86947b5506b8e0e61>

<https://python.hotexamples.com/site/file?hash=0x94d0f36472dbe2f54cdcf3719f04406cd3734608183b719765b49b12942386eb&fullName=capra_lidar-master/laser_scan_merger.py&project=clubcapra/Ibex>

[ERROR] [1648931021.762483479]: [registerPublisher] Failed to contact master at [localhost:11311]. Retrying…

**export** ROS\_MASTER\_URI=[http:*//master:11311*](http://master:11311)

[*https://lightwarelidar.com/products/sf40-c-100-m*](https://lightwarelidar.com/products/sf40-c-100-m)

*~/path\_to\_your\_workspace/devel/setup.bash*

*rosrun lightware\_sf40\_ros sf40\_node /dev/ttyUSB0 115200 ?TM,360,0*

*terminate called after throwing an instance of 'serial::IOException'*

*what(): IO Exception (13): Permission denied, file /home/paul/catkin\_ws/src/serial-main/src/impl/unix.cc, line 151.*

*Aborted (core dumped)*

*[ERROR] [1648936678.063308811]: Error 2 opening : No such file or directory*

*sudo chown paul /dev/ttyUSB0*

*git clone* [*https://bitbucket.org/castacks/sf30\_node.git*](https://bitbucket.org/castacks/sf30_node.git)

*roslaunch sf30\_node sf30.launch*

*http://wiki.ros.org/sf30\_node*

[*https://hackersgrid.com/2017/08/how-to-install-and-setup-linux-in-orange-pi-r1.html*](https://hackersgrid.com/2017/08/how-to-install-and-setup-linux-in-orange-pi-r1.html)

[*https://github.com/LightWare-Optoelectronics/SampleLibrary/blob/master/sf30\_arduino\_serial/sf30\_arduino\_serial.in*](https://github.com/LightWare-Optoelectronics/SampleLibrary/blob/master/sf30_arduino_serial/sf30_arduino_serial.in)

*#!/usr/bin/env python*

[*2*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_2) *# license removed for brevity*

[*3*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_3) *import rospy*

[*4*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_4) *from std\_msgs.msg import String*

[*5*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_5)

[*6*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_6) *def talker():*

[*7*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_7) *pub = rospy.Publisher('chatter', String, queue\_size=10)*

[*8*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_8) *rospy.init\_node('talker', anonymous=True)*

[*9*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_9) *rate = rospy.Rate(10) # 10hz*

[*10*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_10) *while not rospy.is\_shutdown():*

[*11*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_11) *hello\_str = "hello world %s" % rospy.get\_time()*

[*12*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_12) *rospy.loginfo(hello\_str)*

[*13*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_13) *pub.publish(hello\_str)*

[*14*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_14) *rate.sleep()*

[*15*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_15)

[*16*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_16) *if \_\_name\_\_ == '\_\_main\_\_':*

[*17*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_17) *try:*

[*18*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_18) *talker()*

[*19*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_19) *except rospy.ROSInterruptException:*

[*20*](http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29#rospy_tutorials.2FTutorials.2FWritingPublisherSubscriber.CA-c82832e0d612370fe9886563f0b7f5433f6caee1_20) *pass*

[*http://cobecoballes-robotics.blogspot.com/2018/08/laserscan-ros-arduino.html*](http://cobecoballes-robotics.blogspot.com/2018/08/laserscan-ros-arduino.html)

*scan.header.frame\_id = 'laser'*

*scan.header.stamp = get\_most\_recent\_timestamp(rf, sg)*

*scan.angle\_min = angle\_min*

*scan.angle\_max = angle\_max*

*scan.angle\_increment = angle\_increment*

*scan.scan\_time = time.time() - last\_scan\_time*

*scan.time\_increment = scan.scan\_time / 541*

*scan.range\_min = rf.range\_min*

*scan.range\_max = rf.range\_max*

*scan.ranges = rf.ranges*

*for i in range(180\*2):*

*if sg.ranges[i] < scan.ranges[90 + i] or scan.ranges[90 + i] == 0:*

*scan.ranges[90 + i] = sg.ranges[i]*