

**Note:** This tutorial assumes that you have completed the previous tutorials: Arduino IDE Setup (/rosserial\_arduino/Tutorials/Arduino%20IDE%20Setup).

💡 It is appreciated that problems/questions regarding this tutorial are asked on [answers.ros.org](http://answers.ros.org) (http://answers.ros.org). Don't forget to include in your question the link to this page, versions of your OS & ROS, and also add appropriate tags.

# 1. Hello World (example publisher)

**Description:** This tutorial shows step by step how to create a publisher using roscpp.

**Tutorial Level:** BEGINNER

**Next Tutorial:** Blink (example subscriber) (/rosserial\_arduino/Tutorials/Blink)

electric	fuerte	groovy	hydro	indigo	jade
----------	--------	--------	-------	--------	------

**Tabla de Contenidos**

- 1. Hello World: Creating a Publisher
  - 1. The Code
  - 2. The Code Explained
  - 3. Uploading the Code
  - 4. Running the Code
- 2. Further Reading

## 2. Hello World: Creating a Publisher

### 2.1 The Code

We'll start our exploration into roscpp by creating a "hello world" program for our Arduino. (Note: the Arduino community often calls source code for programs a "sketch", we will use the same convention below). If you have followed the Arduino IDE Setup (/rosserial\_arduino/Tutorials/Arduino%20IDE%20Setup) tutorial, you'll be able to open the sketch below by choosing `ros_lib` -> `HelloWorld` from the Arduino examples menu.

This should open the following code in your IDE:

```
[des]activar nros. de línea
```

```
1 /*
2  * roserial Publisher Example
3  * Prints "hello world!"
4  */
5
6 #include <ros.h>
7 #include <std_msgs/String.h>
8
9 ros::NodeHandle nh;
10
11 std_msgs::String str_msg;
12 ros::Publisher chatter("chatter", &str_msg);
13
14 char hello[13] = "hello world!";
15
16 void setup()
17 {
18   nh.initNode();
19   nh.advertise(chatter);
20 }
21
22 void loop()
23 {
24   str_msg.data = hello;
25   chatter.publish( &str_msg );
26   nh.spinOnce();
27   delay(1000);
28 }
```

## 2.2 The Code Explained

Now, let's break the code down.

[des]activar nros. de línea

```
6 #include <ros.h>
7 #include <std_msgs/String.h>
8
```

As a part of every ROS Arduino program, you need to include the `ros.h` header file and header files for any messages that you will be using.

[des]activar nros. de línea

```
9 ros::NodeHandle nh;
```

Next, we need to instantiate the node handle, which allows our program to create publishers and subscribers. The node handle also takes care of serial port communications.

[des]activar nros. de línea

```
11 std_msgs::String str_msg;
12 ros::Publisher chatter("chatter", &str_msg);
```

We need to instantiate the publishers and subscribers that we will be using. Here we instantiate a Publisher with a topic name of "chatter". The second parameter to Publisher is a reference to the message instance to be used for publishing.

[des]activar nros. de línea

```
16 void setup()
17 {
18   nh.initNode();
19   nh.advertise(chatter);
20 }
```

In the Arduino setup function you then need to initialize your ROS node handle, advertise any topics being published, and subscribe to any topics you wish to listen to.

[des]activar nros. de línea

```
22 void loop()
23 {
24   str_msg.data = hello;
25   chatter.publish( &str_msg );
26   nh.spinOnce();
27   delay(1000);
28 }
```

Finally, in the loop function, the node publishes "Hello World" and calls `ros::spinOnce()` where all of the ROS communication callbacks are handled.

## 2.3 Uploading the Code

To upload the code to your Arduino, use the upload function within the Arduino IDE. This is no different from uploading any other sketch.

## 2.4 Running the Code

Now, launch the roscore (/roscore) in a new terminal window:

```
roscore
```

Next, run the roserial client application that forwards your Arduino messages to the rest of ROS. Make sure to use the correct serial port:

```
roslaunch roserial_arduino serial_node.py _port:=/dev/ttyUSB0
```

Finally, watch the greetings come in from your Arduino by launching a new terminal window and entering :

```
rostopic echo chatter
```

### 3. Further Reading

Please see [rosterial/Overview \(/rosterial/Overview/Publishers%20and%20Subscribers\)](/rosterial/Overview/Publishers%20and%20Subscribers) for more information on publishers and subscribers. Also see [limitations \(/rosterial/Overview/Limitations\)](/rosterial/Overview/Limitations) for information about more complex data types.

Except where

otherwise

Wiki: [rosterial\\_arduino/Tutorials>Hello World](#) (última edición 2014-09-22 16:29:23 efectuada por [AustinHendrix \(/AustinHendrix\)](#))

noted, the

ROS wiki is licensed under the

[Creative Commons Attribution 3.0 \(http://creativecommons.org/licenses/by/3.0/\)](http://creativecommons.org/licenses/by/3.0/) | Find us on [Google+](https://plus.google.com/113789706402978299308)

Brought to you by:  Open Source Robotics Foundation

(<http://www.osrfoundation.org>)