**Esempio di Publisher**

Codice

#include <ros.h>

[7](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_7) #include <std\_msgs/String.h>

[8](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_8)

[9](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_9) ros::NodeHandle nh;

[10](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_10)

[11](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_11) std\_msgs::String str\_msg;

[12](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_12) ros::Publisher chatter("chatter", &str\_msg);

[13](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_13)

[14](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_14) char hello[13] = "hello world!";

[15](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_15)

[16](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_16) void setup()

[17](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_17) {

[18](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_18) nh.initNode();

[19](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_19) nh.advertise(chatter);

[20](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_20) }

[21](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_21)

[22](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_22) void loop()

[23](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_23) {

[24](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_24) str\_msg.data = hello;

[25](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_25) chatter.publish( &str\_msg );

[26](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_26) nh.spinOnce();

[27](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_27) delay(1000);

[28](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-f52c3186d145d0ab359f23e0b05b5ea7cee62c79_28) }

Andiamo ad esaminarlo

#include <ros.h>

[7](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-5826d3601828e55baddc97302302a2bbdf6a5f62_7) #include <std\_msgs/String.h>

[8](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-5826d3601828e55baddc97302302a2bbdf6a5f62_8)

E’ una parte che deve essre sempre presente in ogni programma ROS.

[9](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-0931df68a0576b0231e71d81c449487199e1dd1f_9) ros::NodeHandle nh;

Instanzia il nodo principale che ci consente di creare publishers e subscribers.

[11](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-6ce4c2b4ee7163df2d1f0e9cb529d4b8a40ef044_11) std\_msgs::String str\_msg;

[12](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-6ce4c2b4ee7163df2d1f0e9cb529d4b8a40ef044_12) ros::Publisher chatter("chatter", &str\_msg);

Queste due righe di codice servono per instanziare publisher e subscribers con un topic di nome “Chatter”

[16](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-96f01e519593a568c0a728d452aa9670d4b5a2ce_16) void setup()

[17](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-96f01e519593a568c0a728d452aa9670d4b5a2ce_17) {

[18](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-96f01e519593a568c0a728d452aa9670d4b5a2ce_18) nh.initNode();

[19](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-96f01e519593a568c0a728d452aa9670d4b5a2ce_19) nh.advertise(chatter);

[20](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-96f01e519593a568c0a728d452aa9670d4b5a2ce_20) }

Serve ad inizializzare il nodo ROS, che avvertirà la pubblicazione e la sottoscrizione ad un determinato Topic

[22](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-afeb2edcb33158d9707af3fc02fbdf6bd1bf878f_22) void loop()

[23](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-afeb2edcb33158d9707af3fc02fbdf6bd1bf878f_23) {

[24](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-afeb2edcb33158d9707af3fc02fbdf6bd1bf878f_24) str\_msg.data = hello;

[25](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-afeb2edcb33158d9707af3fc02fbdf6bd1bf878f_25) chatter.publish( &str\_msg );

[26](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-afeb2edcb33158d9707af3fc02fbdf6bd1bf878f_26) nh.spinOnce();

[27](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-afeb2edcb33158d9707af3fc02fbdf6bd1bf878f_27) delay(1000);

[28](http://wiki.ros.org/rosserial_arduino/Tutorials/Hello%20World#CA-afeb2edcb33158d9707af3fc02fbdf6bd1bf878f_28) }

In questo pezzo di codice, il nodo che abbiamo recentemente inizializzato pubblica “Hello world” sul topic e viene richiamata la funzione nh.spinOnce(); che permette ci permette di gestire tutte le chiamate.

**Esempio di Subscribers**

Codice

[6](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_6) #include <ros.h>

[7](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_7) #include <std\_msgs/Empty.h>

[8](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_8)

[9](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_9) ros::NodeHandle nh;

[10](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_10)

[11](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_11) void messageCb( const std\_msgs::Empty& toggle\_msg){

[12](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_12) digitalWrite(13, HIGH-digitalRead(13)); // blink the led

[13](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_13) }

[14](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_14)

[15](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_15) ros::Subscriber<std\_msgs::Empty> sub("toggle\_led", &messageCb );

[16](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_16)

[17](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_17) void setup()

[18](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_18) {

[19](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_19) pinMode(13, OUTPUT);

[20](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_20) nh.initNode();

[21](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_21) nh.subscribe(sub);

[22](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_22) }

[23](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_23)

[24](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_24) void loop()

[25](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_25) {

[26](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_26) nh.spinOnce();

[27](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_27) delay(1);

[28](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-7f48e62c93848c28c1e367f430d0051ad1546342_28) }

Andiamo ad esaminarlo

[6](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-4624f3d6209a34554955fb46c11c585bb0e12a56_6) #include <ros.h>

[7](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-4624f3d6209a34554955fb46c11c585bb0e12a56_7) #include <std\_msgs/Empty.h>

Andiamo ad includere la libreria Ros e un header per i messaggi.

[Toggle line numbers](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink)

[9](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-0931df68a0576b0231e71d81c449487199e1dd1f_9) ros::NodeHandle nh;

Instanziamo il nodo che ci permette di creare publishers e subscribers.

[11](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-d2be4794fdc0cb75c46bdbb0a2bef854f261056d_11) void messageCb( const std\_msgs::Empty& toggle\_msg){

[12](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-d2be4794fdc0cb75c46bdbb0a2bef854f261056d_12) digitalWrite(13, HIGH-digitalRead(13)); // blink the led

[13](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-d2be4794fdc0cb75c46bdbb0a2bef854f261056d_13) }

Creiamo la funzione callback per i subscibers.La funzione di callback deve contenere una constante che fa riferimento all’argomento.

Il tipo del messaggio è: std\_msgs::Empty mentre il nome è: toggle\_msg

[15](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-15b3bf02ab3052e2642f972a6c600782a9196ca4_15) ros::Subscriber<std\_msgs::Empty> sub("toggle\_led", &messageCb );

Arrivato a questo punto dobbiamo instanziare il publishers e il subscribers. Instanziamo il subscribers con un topic di nome: toggle\_led e tipo: std\_msgs:: Empty

[17](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-6ae9cd358bd0ac5073c1712539920fc931f10fb1_17) void setup()

[18](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-6ae9cd358bd0ac5073c1712539920fc931f10fb1_18) {

[19](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-6ae9cd358bd0ac5073c1712539920fc931f10fb1_19) pinMode(13, OUTPUT);

[20](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-6ae9cd358bd0ac5073c1712539920fc931f10fb1_20) nh.initNode();

[21](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-6ae9cd358bd0ac5073c1712539920fc931f10fb1_21) nh.subscribe(sub);

[22](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-6ae9cd358bd0ac5073c1712539920fc931f10fb1_22) }

Nel void setup dobbiamo inizializzare il nodo Ros e avvertire il topic che verra pubblicizzato inoltre dobbiamo avvertire il subscribers di inscriversi al topic.

[24](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-3c13d57378bfddd71f741800140388b782c15c8c_24) void loop()

[25](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-3c13d57378bfddd71f741800140388b782c15c8c_25) {

[26](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-3c13d57378bfddd71f741800140388b782c15c8c_26) nh.spinOnce();

[27](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-3c13d57378bfddd71f741800140388b782c15c8c_27) delay(1);

[28](http://wiki.ros.org/rosserial_arduino/Tutorials/Blink#CA-3c13d57378bfddd71f741800140388b782c15c8c_28) }

Infine nel void loop andiamo a richiamare la funzione nh.spinOnce(); dove tutte le comunicazioni Ros vengono richiamate.

**Log dei Nodi**

Codice

[1](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_1) #include <ros.h>

[2](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_2) ros::NodeHandle nh;

[3](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_3)

[4](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_4) void setup()

[5](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_5) {

[6](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_6) nh.initNode();

[7](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_7) }

[8](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_8)

[9](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_9) void loop()

[10](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_10) {

[11](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_11) //wait until you are actually connected

[12](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_12) while (!nh.connected())

[13](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_13) {

[14](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_14) nh.spinOnce();

[15](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_15) }

[16](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_16)

[17](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_17) //Now you can send all of the statements you want

[18](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_18) //to be logged at the appropriate verbosity level

[19](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_19) nh.logdebug("Debug Statement");

[20](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_20) nh.loginfo("Program info");

[21](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_21) nh.logwarn("Warnings.");

[22](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_22) nh.logerror("Errors..");

[23](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_23) nh.logfatal("Fatalities!");

[24](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_24) delay(5000);

[25](http://wiki.ros.org/rosserial_arduino/Tutorials/Logging#CA-4f8ee899d31cc2666337567dbef4e1137a83052b_25) }

Dobbiamo fare attenzione in quanto se andassimo a leggere le informazioni senza nessun Hardware “collegato” a Ros verrà visualizzato un messaggio con il contenuto di : Informazioni perse.

Dobbiamo inoltre ricordarci che