

```
// %Tag(FULLTEXT)%
// %Tag(ROS_HEADER)%
#include "ros/ros.h"
// %EndTag(ROS_HEADER)%
// %Tag(MSG_HEADER)%
#include "std_msgs/String.h"
#include "std_msgs/Empty.h"
#include "std_msgs/Int32.h"
// %EndTag(MSG_HEADER)%

#include <sstream>
#include <iostream>
using namespace std;

/**
 * This tutorial demonstrates simple sending of messages over the ROS system.
 */
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");

    ros::NodeHandle n;

    //create the publishers
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Publisher chatter_pub2 = n.advertise<std_msgs::Int32>("motor1", 1000);
    ros::Publisher chatter_pub3 = n.advertise<std_msgs::Int32>("motor2", 1000);

    ros::Rate loop_rate(10);

    int count = 0;

    //allows for input to direct the motor either up or down
    while (ros::ok())
    {
        std_msgs::String msg;

        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();

        ROS_INFO("%s", msg.data.c_str());

        chatter_pub.publish(msg);

        std_msgs::Int32 myMsg;
        int numFlashes;
        int line;
        cout << "Enter 1 for up, and 2 for down: ";

        cin >> line;
        cin.ignore();
        cout << "Enter number of steps: ";
        cin >> numFlashes;

        myMsg.data = numFlashes;

        switch(line) {

            case 1:
                for (int i = 0; i < 200; i++) {
                    chatter_pub2.publish(myMsg);
                }
                break;
            case 2:
```

```
    chatter_pub3.publish(myMsg);  
    break;  
  
}  
    ros::spinOnce();  
  
    loop_rate.sleep();  
    ++count;  
}  
  
    return 0;  
}  
// %EndTag(FULLTEXT)%
```