

**Dokumentacja z laboratorium nr 1 z przedmiotu ANRO**  
**Rafał Koguciuk**  
**Jakub Wieczorek**

Naszym zadaniem było stworzyć węzeł, który umożliwiłby kierowanie żółwiem symulacyjnym na dowolnych przyciskach z klawiatury. Węzeł ten ma za zadanie publikować wiadomości konkretnego typu na odpowiednim topicu.

Kod programu (publisher):

```
#include "ros/ros.h"
#include "geometry_msgs/Twist.h"
#include <iostream>
#include <stdio.h>
#include <termios.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/time.h>

//funkcje pomocnicze do wczytywania znaku z klawiatury
void changemode(int);
int kbhit(void);

void changemode(int dir)
{
    static struct termios oldt, newt;

    if ( dir == 1 )
    {
        tcgetattr( STDIN_FILENO, &oldt);
        newt = oldt;
        newt.c_lflag &= ~( ICANON | ECHO );
        tcsetattr( STDIN_FILENO, TCSANOW, &newt);
    }
    else
        tcsetattr( STDIN_FILENO, TCSANOW, &oldt);
}

int kbhit (void)
{
    struct timeval tv;
    fd_set rdfs;

    tv.tv_sec = 0;
    tv.tv_usec = 0;

    FD_ZERO(&rdfs);
    FD_SET (STDIN_FILENO, &rdfs);

    select(STDIN_FILENO+1, &rdfs, NULL, NULL, &tv);
    return FD_ISSET(STDIN_FILENO, &rdfs);
}
```

```
}
```

```
//główna funkcja programu
```

```
int main(int argc, char **argv)
{
```

```
    ros::init(argc, argv, "talker");
```

```
    ros::NodeHandle n;
```

```
    //będziemy publikować na topicu o nazwie turtle1/cmd_vel zmienną typu geometry_msgs::Twist
    ros::Publisher chatter_pub = n.advertise<geometry_msgs::Twist>("/turtle1/cmd_vel", 1000);
```

```
    ros::Rate loop_rate(10);
```

```
    unsigned char znak;
```

```
//główna pętla programu
```

```
while (ros::ok())
{
```

```
    //zmienna pomocnicza
```

```
    geometry_msgs::Twist twist;
    changemode(1);
```

```
    //jeśli został wcisnięty jakiś klawisz sprawdzamy czy to był jeden z w,a,s,d i na tej podstawie
    zmieniamy wartość zmiennej pomocniczej.
```

```
while ( kbhit() ){
    znak = getchar();
```

```
    if (znak=='w') {twist.linear.x = 2.0; ROS_INFO("%s", "w");}; //do przodu
```

```
    if (znak=='s') {twist.linear.x = -2.0; ROS_INFO("%s", "s");}; //do tyłu
```

```
    if (znak=='a') {twist.angular.z = 2.0;ROS_INFO("%s", "a");}; //w lewo
```

```
    if (znak=='d') {twist.angular.z = -2.0;ROS_INFO("%s", "d");}; //w prawo
```

```
}
```

```
    changemode(0);
```

```
//publikowanie na topicu wiadomości typu geometry_msgs::Twist
```

```
chatter_pub.publish(twist);
```

```
ros::spinOnce();
```

```
loop_rate.sleep();
```

```
}
```

```
return 0;
```

```
}
```

**Sposób uruchomienia roslauncha, wpisujemy kolejno komendy w terminalu:**

```
roscore                //(Terminal 1.)  
cd ~/catkin_ws         //(Terminal 2.)  
source ./devel/setup.bash  
roslaunch beginner_tutorials turtle_mimic.launch
```