

```
#include <ros.h> //dedicada à comunicação dos pacotes ROS
#include <ros/time.h> //responsável pela sincronização dos
tempos ROS
#include <geometry_msgs/Twist.h> //trata do tipo de mensagem
ROS
```

```
#define IN1 10 //define que o pino 10 receberá o valor da
variável IN1
#define IN2 11 //define que o pino 11 receberá o valor da
variável IN2
#define IN3 12 //define que o pino 12 receberá o valor da
variável IN3
#define IN4 13 //define que o pino 13 receberá o valor da
variável IN4
```

```
void onTwist(const geometry_msgs::Twist& msg)
{
    if(msg.linear.x > 0) //TUPY PARA FRENTE
    {
        digitalWrite(IN1,HIGH); //roda direita para frente ON
        digitalWrite(IN2,LOW); //roda direita para trás OFF
        digitalWrite(IN3,HIGH); //roda esquerda para frente ON
        digitalWrite(IN4,LOW); //roda esquerda para trás OFF
    }
    else if(msg.linear.x < 0) //TUPY PARA TRÁS
    {
        digitalWrite(IN1,LOW); //roda direita para frente OFF
        digitalWrite(IN2,HIGH); //roda direita para trás ON
        digitalWrite(IN3,LOW); //roda esquerda para frente OFF
        digitalWrite(IN4,HIGH); //roda esquerda para trás ON
    }
    else if(msg.angular.z < 0) //TUPY GIRA PARA ESQUERDA
    {
        digitalWrite(IN1,HIGH); //roda direita para frente ON
        digitalWrite(IN2,LOW); //roda direita para trás OFF
        digitalWrite(IN3,LOW); //roda esquerda para frente OFF
        digitalWrite(IN4,HIGH); //roda esquerda para trás ON
    }
    else if(msg.angular.z > 0)//TUPY GIRA PARA DIREITA
    {
        digitalWrite(IN1,LOW); //roda direita para frente OFF
        digitalWrite(IN2,HIGH); //roda direita para trás ON
        digitalWrite(IN3,HIGH); //roda esquerda para frente ON
        digitalWrite(IN4,LOW); //roda esquerda para trás OFF
    }
    else //PARAR TUPY
```

```

    {
        digitalWrite(IN1,LOW); //roda direita para frente OFF
        digitalWrite(IN2,LOW); //roda direita para trás OFF
        digitalWrite(IN3,LOW); //roda esquerda para frente OFF
        digitalWrite(IN4,LOW); //roda esquerda para trás OFF
    }
}
//Inscreve objetos do referentes aos comandos da teleoperação
ros::Subscriber<geometry_msgs::Twist> sub("cmd_vel",onTwist);
//cria um objeto que representa o nó ROS.
ros::NodeHandle nh; //iniciará o nó na placa Arduino.

void setup()    // código de configuração para executar uma
vez:
{
    pinMode(IN1,OUTPUT); //Configura a variável IN1 (pin 10)
como saída digital
    pinMode(IN2,OUTPUT); //Configura a variável IN2 (pin 11)
como saída digital
    pinMode(IN3,OUTPUT); //Configura a variável IN3 (pin 12)
como saída digital
    pinMode(IN4,OUTPUT); //Configura a variável IN4 (pin 13)
como saída digital

    nh.initNode(); //inicia o nó ROS para o processo
    nh.subscribe(sub); //define que a programação irá inscrição
do nó
}

void loop() //código principal para ser executado
repetidamente:
{
    nh.spinOnce(); //Informa ao ROS que uma nova mensagem
chegou.
}

```