Ingeniería mecatrónica

9°B T/M

Tarea 4

ALUMNO

Lozano Ochoa Marco Antonio

Prof. Moran Garabito Carlos Enrique

Asignatura: Dinámica y control de robots

Objetivo:

Controlar el movimiento de un motor paso a paso utilizando ROS y un microcontrolador de 32 bits.

Materiales:

- Computadora con software ROS.
- Motor PaP.
- Fuente de alimentación.
- Microcontrolador FRDM KL25Z.
- Cable dupont para conexiones.
- Driver L298N.

Procedimiento:

1. Se programa la KL25Z mediante mbed con el programa de suscriptor.

2. Se hacen las conexiones del motor con el driver y el microcontrolador.

3. Se ejecuta el nodo principal de ROS en terminal mediante el siguiente comando:

roscore

4. Se dan privilegios de lectura y escritura al puerto donde se tiene conectado el microcontrolador para poder publicar los mensajes:

sudo chmod 666 /dev/ttyACM0

5. Se ejecuta el nodo serial de Python en ROS:

rosrun rosserial_python serial_node.py /dev/ttyACM0

6. Se hace una publicación al nodo suscriptor del microcontrolador para mandar un mensaje de tipo int16:

rostopic pub servo std_msgs/Int16 <numero de pasos del motor> --once

Nota: El programa fue elaborado para aceptar el numero de pasos del 1 al 20 para controlar el sentido de giro antihorario, y con un numero de pasos mayor a 20 se moverá en sentido horario, debido a que los mensajes con valores negativos ROS los interpreta como instrucciones y no como un dato.

Código de programación de la KL25Z como nodo suscriptor:

```
#include "mbed.h"                    //Importación de las librerias de mbed y ROS
#include <ros.h>
#include <std_msgs/Int16.h>

ros::NodeHandle  nh;                 //Nombramiento del nodo a manejar
```

```
DigitalOut IN1(D2);              //Declaración de los pines de control
DigitalOut IN2(D3);
DigitalOut IN3(D4);
DigitalOut IN4(D5);
float stepDelay = 0.050;        //Tiempo para el control de velocidad del motor

void servo_cb( const std_msgs::Int16& cmd_msg)      //Función para la lectura del mensaje
{
   int vrec = cmd_msg.data;                    //Guardar el mensaje en una variable
   if(vrec > 20) {
      vrec = vrec-20;                         //Procedimiento para controlar el sentido de giro del motor
      for (int x = 0; x < vrec; x++) {
         IN1=1;
         IN2=0;
         IN3=0;
         IN4=0;
         wait(stepDelay);
         IN1=0;
         IN2=1;
         IN3=0;
         IN4=0;
         wait(stepDelay);
         IN1=0;
         IN2=0;
         IN3=1;
         IN4=0;
         wait(stepDelay);
         IN1=0;
         IN2=0;
         IN3=0;
         IN4=1;
         wait(stepDelay);
      }
   }
   if(vrec < 20) {                                  //Procedimiento para cambiar el sentido
      for (int x = 0; x < vrec; x++) {
         IN1=0;
         IN2=0;
         IN3=0;
         IN4=1;
         wait(stepDelay);
         IN1=0;
         IN2=0;
         IN3=1;
         IN4=0;
         wait(stepDelay);
         IN1=0;
```

```
        IN2=1;
        IN3=0;
        IN4=0;
        wait(stepDelay);
        IN1=1;
        IN2=0;
        IN3=0;
        IN4=0;
        wait(stepDelay);
      }
   }
}

//El nodo crea un suscriptor llamado servo
ros::Subscriber<std_msgs::Int16> sub("servo", servo_cb);

int main()
{

   nh.initNode();                //Inicialización del nodo
   nh.subscribe(sub);

   while (1) {
      nh.spinOnce();            //El nodo solo se ejecutará una vez al recibir un mensaje
      wait_ms(1);               //Retraso de un segundo
   }
}
```