

UNIVERSIDAD POLITECNICA DE LA ZONA METROPOLITANA DE GUADALAJARA

DINAMICA Y CONTROL DE ROBOTS



TAREA #4

INGENIERIA MECATRONICA 9°B

MAESTRO:

CARLOS ENRIQUE MORAN GARABITO

ALUMNO:

ALEXIS ISRAEL VIORATO ARAMBULA

INTRODUCCION

A continuación, les mostrare el contenido de la tarea numero 4 en la cual comunicamos ros con un microcontrolador que en mi caso fue la Cypress, para manipular un motor a pasos programado en el micro,

OBJETIVO:

Hacer una comunicación serial entre un microcontrolador y el software ros-kinetic.

MATERIALES:

Computadora con Ubuntu.

Software ros.

Software PSoC

Microcontrolador Cypress

Motor a pasos

Driver L298N

CODIGO DE PSoC:

```
#include <ros.h>
#include <std_msgs/Int16.h>
#include "StepperControl.h"

extern ros::NodeHandle nh;

namespace StepperControl {

const uint8_t msec_per_step = 10;

int16_t steps_remaining = 0;
int16_t dato = 0;
int16_t paso = 0;
uint32_t next_time;

void messageCb( const std_msgs::Int16& step_msg){
    steps_remaining = step_msg.data;
}

ros::Subscriber<std_msgs::Int16> sub("stepper_move", &messageCb );

void setup()
{
    nh.subscribe(sub);
    next_time = SysTimer::millis();
}
```

```

void loop()
{
  if ((int32_t)(SysTimer::millis()-next_time)>0) {
    next_time += msec_per_step;
    if (steps_remaining > 0) {
      dato = steps_remaining;
      dato = (dato*1.422222222);
    }
    if (dato > paso) {
      //paso1
      IN4_Write(1);
      IN2_Write(0);
      IN3_Write(0);
      IN1_Write(1);
      CyDelay(2);
      //paso2
      IN4_Write(0);
      IN2_Write(0);
      IN3_Write(0);
      IN1_Write(1);
      CyDelay(2);
      //paso3
      IN4_Write(0);
      IN2_Write(0);
      IN3_Write(1);
      IN1_Write(1);
      CyDelay(2);
      //paso4
      IN4_Write(0);
      IN2_Write(0);
      IN3_Write(1);
      IN1_Write(0);
      CyDelay(2);
      //paso5
      IN4_Write(0);
      IN2_Write(1);
      IN3_Write(1);
      IN1_Write(0);
      CyDelay(2);
      //paso6
      IN4_Write(0);
      IN2_Write(1);
      IN3_Write(0);
      IN1_Write(0);
      CyDelay(2);
      //paso7
      IN4_Write(1);
      IN2_Write(1);
      IN3_Write(0);
      IN1_Write(0);
      CyDelay(2);
      //paso8
      IN4_Write(1);
      IN2_Write(0);
      IN3_Write(0);
      IN1_Write(0);
      CyDelay(2);
    }
  }
}

```

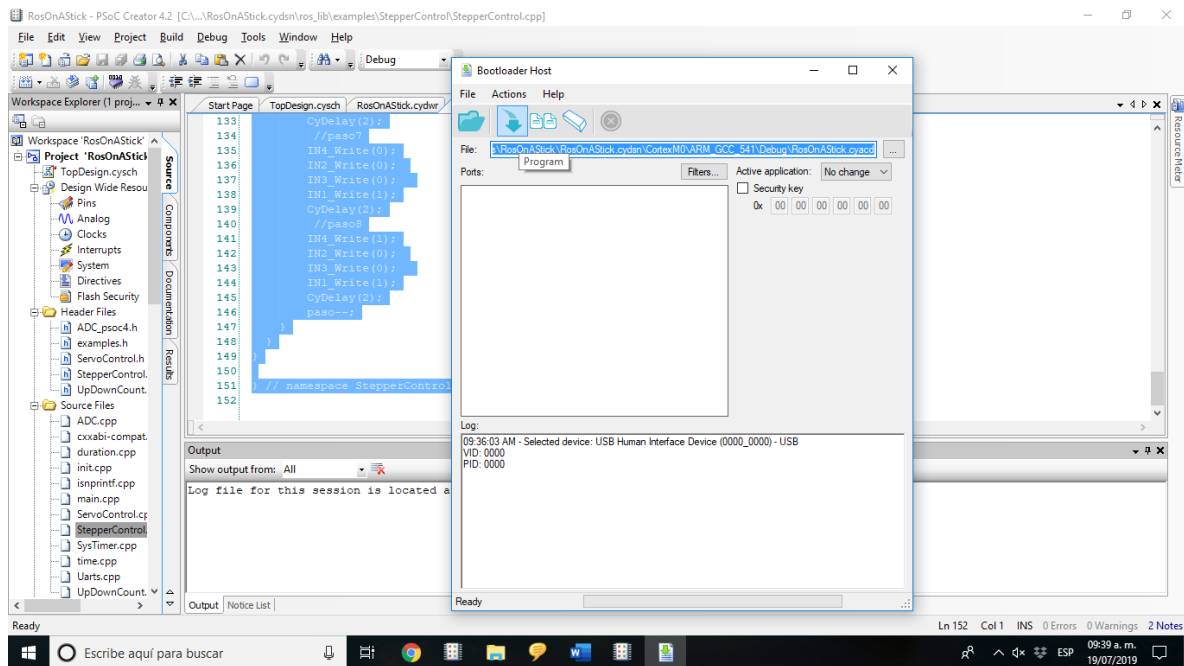
```

        paso++;
    }
    if (dato < paso) {
        //paso1
        IN4_Write(1);
        IN2_Write(0);
        IN3_Write(0);
        IN1_Write(0);
        CyDelay(2);
        //paso2
        IN4_Write(1);
        IN2_Write(1);
        IN3_Write(0);
        IN1_Write(0);
        CyDelay(2);
        //paso3
        IN4_Write(0);
        IN2_Write(1);
        IN3_Write(0);
        IN1_Write(0);
        CyDelay(2);
        //paso4
        IN4_Write(0);
        IN2_Write(1);
        IN3_Write(1);
        IN1_Write(0);
        CyDelay(2);
        //paso5
        IN4_Write(0);
        IN2_Write(0);
        IN3_Write(1);
        IN1_Write(0);
        CyDelay(2);
        //paso6
        IN4_Write(0);
        IN2_Write(0);
        IN3_Write(1);
        IN1_Write(1);
        CyDelay(2);
        //paso7
        IN4_Write(0);
        IN2_Write(0);
        IN3_Write(0);
        IN1_Write(1);
        CyDelay(2);
        //paso8
        IN4_Write(1);
        IN2_Write(0);
        IN3_Write(0);
        IN1_Write(1);
        CyDelay(2);
        paso--;
    }
}
}
}

```

DESARROLLO

1. Primero tenemos que cargarle el programa que viene en el archivo que nos paso el profesor a nuestra cypress, para esto tenemos que compilar primero nuestro programa en PSoC y conectar a nuestra cypress a la computadora en modo Bootloadable. Después nos vamos a la parte de arriba donde dice Tools y le damos click y vamos a seleccionar la parte que dice Bootloader Host y vamos a seleccionar el cyacd correspondiente a nuestro programa y ya solo le damos en la flecha azul de la parte superior izquierda para descargarle el programa a la tarjeta.



2. En este paso vamos a hacer la conexión de la cypress con el motor a pasos la cual es muy sencilla porque solamente es conectar los pines que están declarados a nuestra tarjeta y de la tarjeta al driver para el motor a pasos.
3. Ahora vamos a abrir ubuntu y vamos a correr en la terminal primero el roscore.
4. Después vamos a abrir otra ventana en la terminal y habilitamos el puerto con el siguiente comando: `roslaunch rosserial_python serial_node.py /dev/ttyUSB0`
5. Por ultimo vamos a abrir otra ventana en la terminal y ahí manipularemos el motor con otro comando y en este pondremos los grados que haremos girar el motor a pasos como por ejemplo: `rostopic pub toggle_led std_msgs/Empty 180--once`

CONCLUSION

En conclusión, esta práctica fue algo fácil porque el código y todo el proceso que se tiene que hacer para poder comunicar la Cypress con ROS ya estaba en el archivo que nos pasaron, solamente tuvimos que cargar el programa a la tarjeta y poner los comandos en ROS.