Control

tarea 7

Ricardo Martínez Jacinto

[Año]

Después de haber leído el documento proporcionado por el profesor (Ros con falditas) proceder con su realización

Objetivo:

Obtener una lectura de una tarjeta microcontroladora utilizando Ros

Materiales utilizados en esta práctica:

* Arduino y software para programarlo
* Tarjeta Cypress 049-42XX y software para programarlo
* En este caso se trabajará con una máquina virtual con Ubuntu 16
* Potenciómetro

Código en Arduino, pin utilizado para el potenciómetro A0 en Arduino (físico)

***#include <ros.h>***

***#include <std\_msgs/Float32.h>***

***ros::NodeHandle nh;***

***std\_msgs::Float32 temp\_msg;***

***ros::Publisher pub\_temp("pot", &temp\_msg);***

***int analogPin=A0;***

***int valor;***

***int posicion;***

***void setup()***

***{***

***nh.initNode();***

***nh.advertise(pub\_temp);***

***}***

***void loop()***

***{***

***valor=analogRead(analogPin);***

***temp\_msg.data = valor;***

***pub\_temp.publish(&temp\_msg);***

***nh.spinOnce();***

***delay(1000);***

***}***

Codigo en PSoC Creator, pin utilizado para el potenciómetro: pin 1 (físico)

***/\****

***\* rosserial Subscriber Example***

***\* Blinks an LED on callback***

***\*/***

***extern "C" {***

***#include "project.h"***

***}***

***#include <ros.h>***

***#include <std\_msgs/Empty.h>***

***extern ros::NodeHandle nh;***

***namespace Blink {***

***uint8\_t ledOn;***

***void messageCb( const std\_msgs::Empty& toggle\_msg){***

***ledOn = 1-ledOn;***

***}***

***ros::Subscriber<std\_msgs::Empty> sub("toggle\_led", &messageCb );***

***void setup()***

***{***

***ledOn = 0;***

***P1\_6\_Write(ledOn);***

***nh.subscribe(sub);***

***}***

***void loop()***

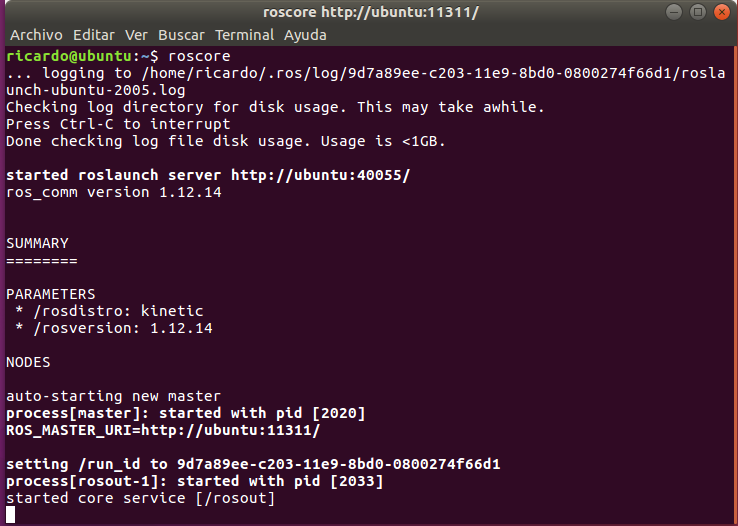
***{***

***P1\_6\_Write(ledOn);***

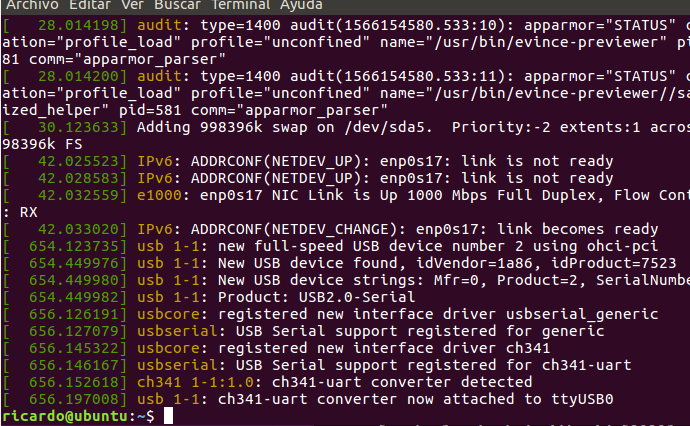
***}***

***} // namespace Blink***

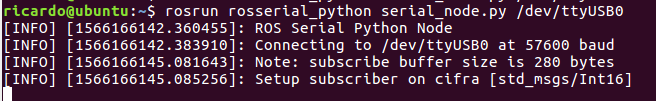
Inicialización de Ros en Ubuntu “roscore”



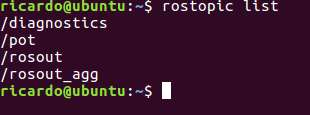
Comprobamos que el dispositivo se haya conectado correctamente con el comando “dmesg”



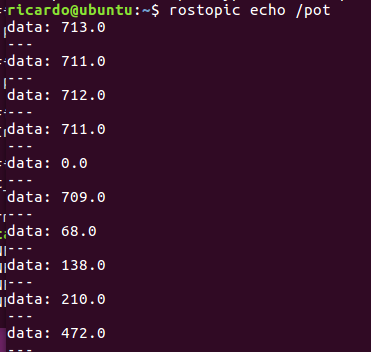
Iniciamos la comunicación con el comando “rosrun rosserial\_pythn serial\_node.py /dev/\*ubicación de nuestra tarjeta\*



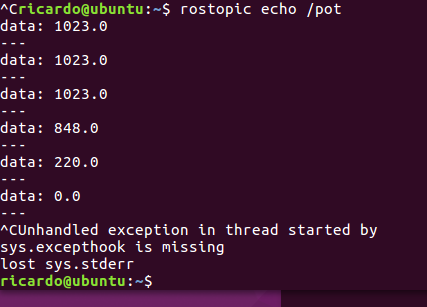
Con el código rostopic list podemos ver que codigos tiene nuestra tarjeta



Con el código “rostopic echo /pot” obtendremos la lectura del potenciómetro



Variando el potenciómetro podremos obtener el valor de 0 a 1023



**Conclusión**

Con esta práctica logramos la comunicación entre ros y la tarjeta microcontroladora, lo cual nos permitirá saber la ubicación de un objeto en el espacio, lo cual puede ser aplicado para una ubicación espacial de algún eslabón de un brazo robótico o de rangos de voltaje que pueden ser utilizados para encadenarse a actividades automatizadas, como encender leds, activar alarmas, o cualquier otra cosa que deseemos.