|  |  |
| --- | --- |
| ***Ingeniería mecatrónica***  9°B T/M | Ramirez Arenas Juan Alberto  **Prof.** Moran Garabito Carlos Enrique  **Asignatura:** Dinámica y control de robots  EV\_4\_1\_ Control |

Objetivo:

Comunicación serial entre Arduino y freescale con ROS.

Material:

Arduino

KL25Z

Ubuntu con ROS

Introducción:

La comunicación serial con ROS mediante un micro controlador se lleva a cabo con la librería rosserial está instalada en Ubuntu junto con ROS, para los microcontroladores las librerías se instalan en el IDE del micro (ejemplo Arduino) en MBED estas librerías se importa en el programa que se está haciendo.

Procedimiento:

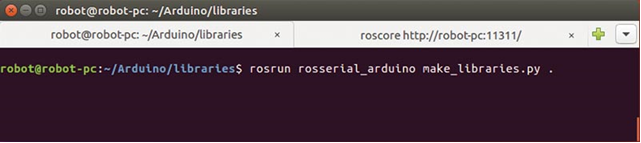
1. Se instala la librería rosserial con el siguiente comando:

*sudo apt-get install ros-kinectic-rosserial*

nota: la instalación de la librería dependerá de la distribución de Ros que este usando (kinetic, indigo o alguna otra).

1. Una vez instalada la librería se instala el IDE de Arduino y una vez instalado, se configura el rosserial-client para la comunicación serial entre el micro y ROS.

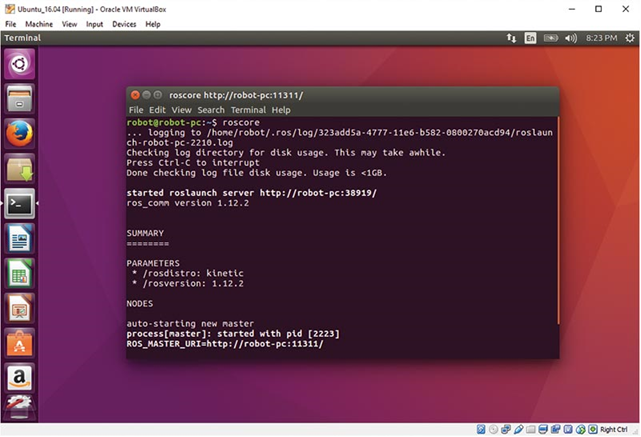
Dentro de la carpeta donde están las librerías de Arduino se ejecuta el siguiente comando que creara la ros\_lib

*rosrun rosserial\_arduino make\_libraries.py*

Para el microcontrolador KL25Z no es necesario instalar librerías ya que al momento de hacer el programa y poner la librería rosserial MBED te marca el importar la librería a tu programa.

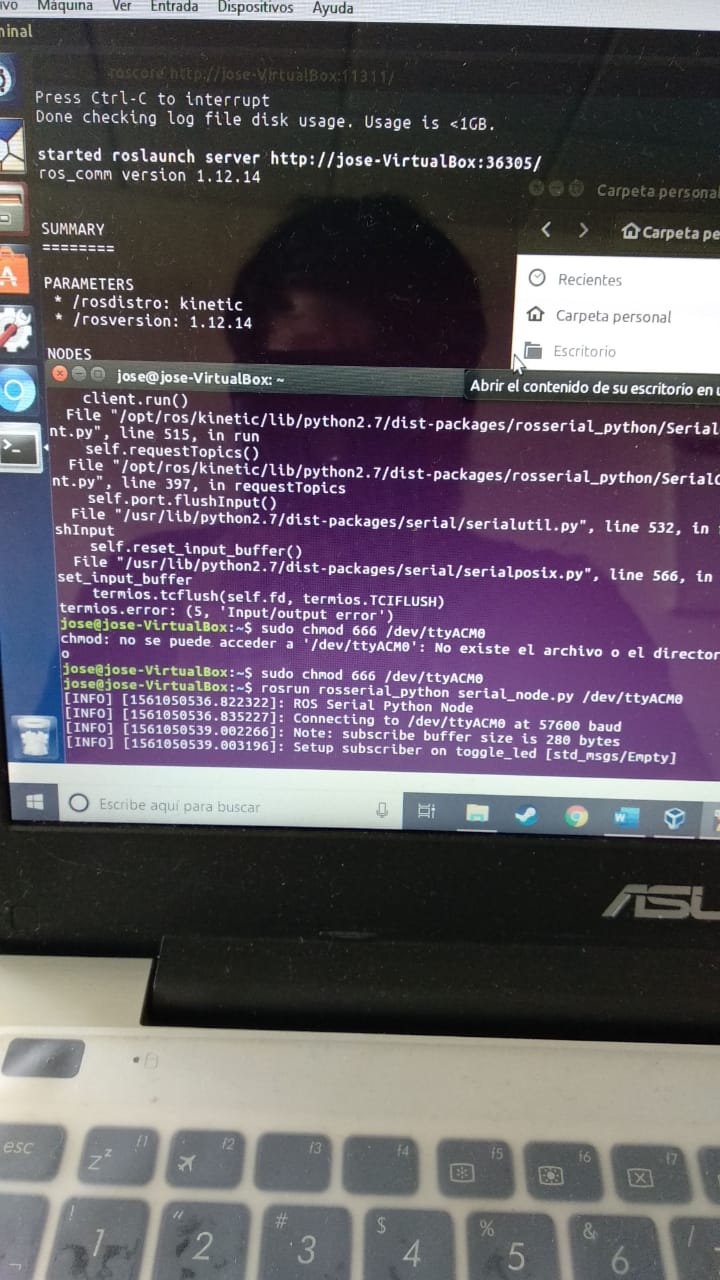
1. Una ves importadas e instaladas las librerías de los dos micros, se ejecuta ROS con el siguiente comando:

*roscore*

*z*

1. Una vez ejecutado ROS se conecta el micro con el programa ya cargado y se ejecuta el siguiente comando en otro terminal:

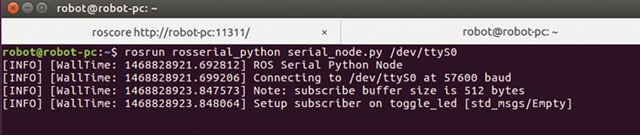
*sudo chmod 666 /dev/ttyACM0 o ttyS0*



Este comando permite que se ejecute el puerto como super usuario permitiendo la comunicación serial entre ROS y el microcontrolador.

1. Después de usar el comando anterior se ejecuta el comando de conexión serial entre el micro y ROS

*Rosrun rosserial\_python serial\_node.py /dev /ttyAMC0*

**

1. Después de que se haya ejecutado el comando anterior y sin ningún error en una nueva terminal de ejecuta el comando el cual hará la visualización y comunicación entre el micro y ROS. Los comandos pueden variar dependiendo los datos que enviara el micro a ROS.

Ejemplo del comando para encender y apagar el led de Arduino.

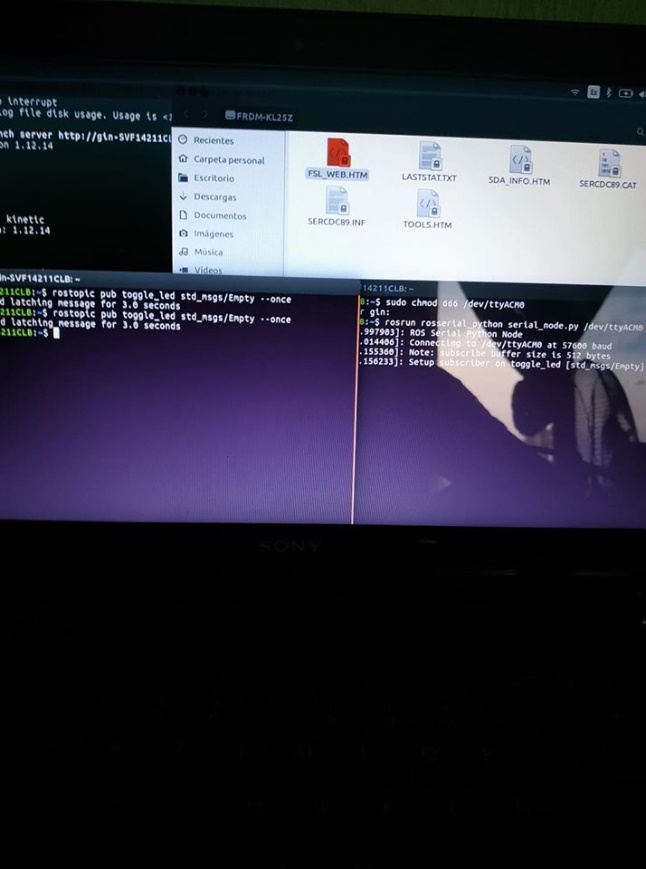
*rostopic pub toggle\_led std\_msgs/Empty --once*

Resultados:



Led encendido

Led apagado



Comando para encender y apagar el led

Conclusión:

Ros es un sistema orientado para el trabajo con robots y esta tarea tuvo como resultado familiarizarme mas con ros y tener mas clara la idea de como usarlo y hacer la comunicación con un microcontrolador y ros por medio de una computadora o raspberry, aunque para gusto personal en computadora es más cómodo por los recursos del equipo.