



Ingeniería mecatrónica
9°B T/M

Navarro Cervantes José

Prof. Moran Garabito Carlos Enrique

Asignatura: Dinámica y control de robots

EV_4_1_ Control

Objetivo:

Comunicación serial entre aduino y freescale con ROS.

Material:

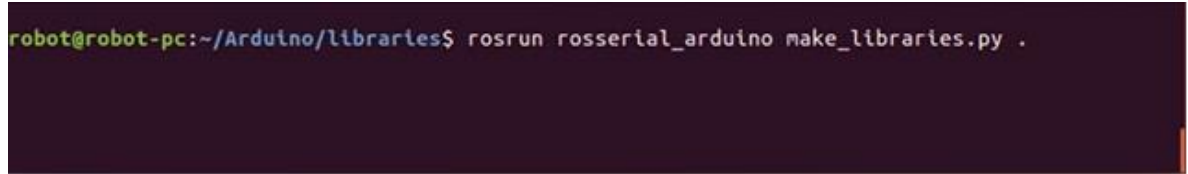
Arduino
KL25Z
Ubuntu con ROS

Introducción:

La comunicación serial con ROS mediante un micro controlador se lleva a cabo con la librería *rosserial* esta 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:

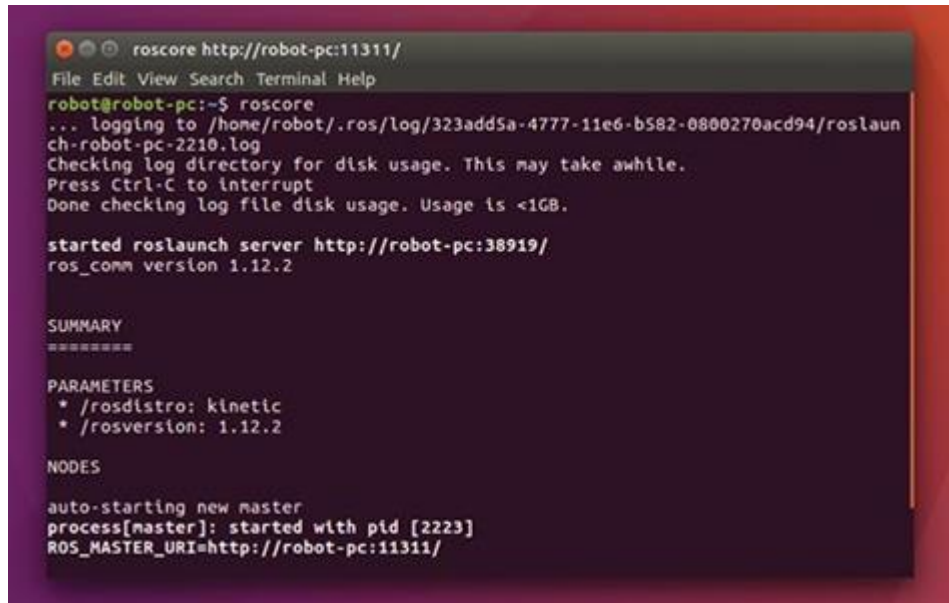
- 1- Se instala la librería *rosserial* con el siguiente comando:
sudo apt-get install ros-kinectic-rosserial
nota: la instalación de la librería dependerá de la distribución de Ros que este usando (kinetic, indigo o alguna otra).
- 2- Una vez instalada la librería se instala el IDE de Arduino y una vez instalado, se cunfigura 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*
roscpp rosserial_arduino make_libraries.py



```
robot@robot-pc:~/Arduino/libraries$ roscpp 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.

- 3- Una vez importadas e instaladas las librerías de los dos micros, se ejecuta ROS con el siguiente comando:
roscpp

A terminal window titled 'roscore http://robot-pc:11311/' showing the process of starting a ROS master. The user runs 'roscore' and the system logs in, checks disk usage, and starts the 'roslaunch server'. It then displays a summary of parameters like 'roslistro: kinetic' and 'rosversion: 1.12.2', and confirms the master is starting with PID 2223.

```
roscore http://robot-pc:11311/
File Edit View Search Terminal Help
robot@robot-pc:~$ roscore
... logging to /home/robot/.ros/log/323add5a-4777-11e6-b582-0800270acd94/roslaun
ch-robot-pc-2210.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://robot-pc:38919/
ros_comm version 1.12.2

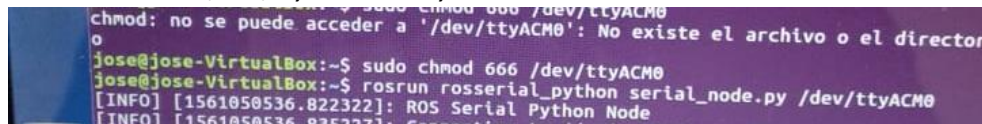
SUMMARY
=====
PARAMETERS
* /roslistro: kinetic
* /rosversion: 1.12.2

NODES

auto-starting new master
process[master]: started with pid [2223]
ROS_MASTER_URI=http://robot-pc:11311/
```

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

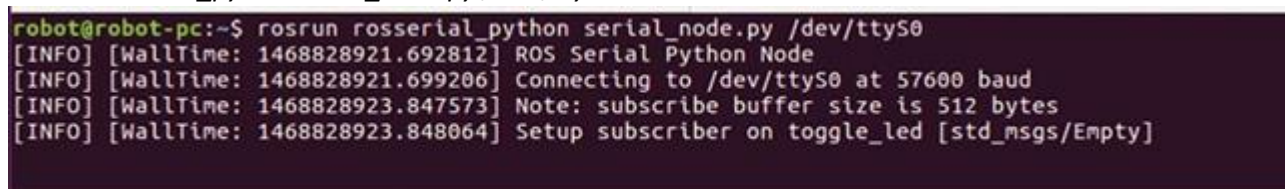
sudo chmod 666 /dev/ttyACM0 or ttyS0

A small terminal snippet showing an error when running 'sudo chmod 666 /dev/ttyACM0'. The error message is 'chmod: no se puede acceder a '/dev/ttyACM0': No existe el archivo o el directorio'.

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

- 5- Después de usar el comando anterior se ejecuta el comando de conexión serial entre el micro y ROS

Rosrun roserial_python serial_node.py /dev/ttyACM0

A terminal window showing the execution of 'rosrun roserial_python serial_node.py /dev/ttyS0'. It displays several INFO messages: 'ROS Serial Python Node', 'Connecting to /dev/ttyS0 at 57600 baud', 'Note: subscribe buffer size is 512 bytes', and 'Setup subscriber on toggle_led [std_msgs/Empty]'.

- 6- 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:

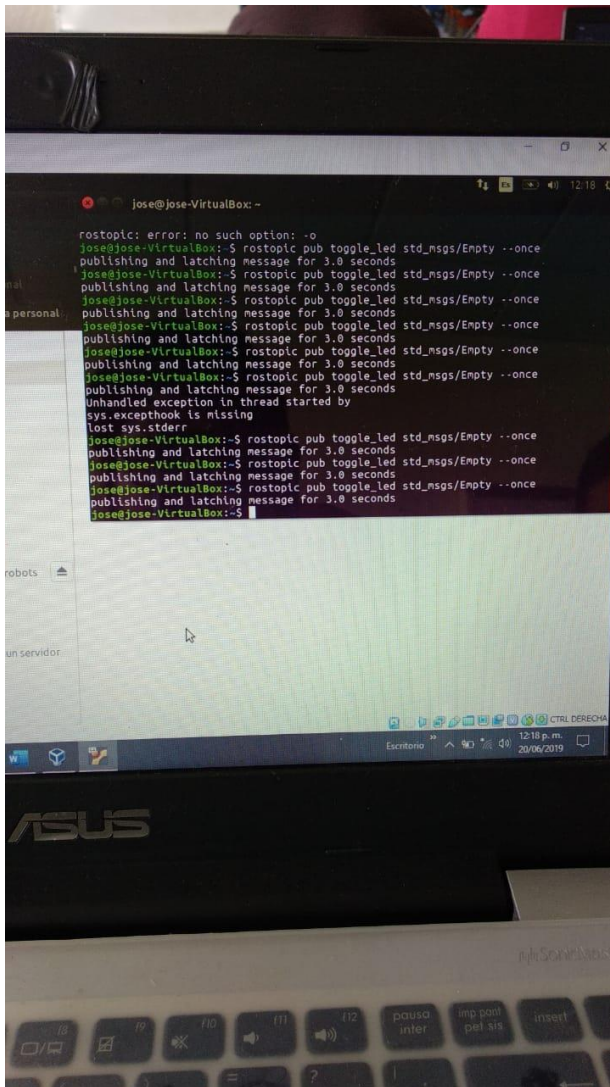


Led apagado



Led encendido

Comando para encender y apagar el led



```
jose@jose-VirtualBox: ~
rostopic: error: no such option: -o
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
Unhandled exception in thread started by
sys.excepthook is missing
lost sys.stderr
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$ rostopic pub toggle_led std_msgs/Empty --once
publishing and latching message for 3.0 seconds
jose@jose-VirtualBox:~$
```

Conclusión:

La comunicación serial con ROS depende de los mensajes o datos que quieras enviar para que ingreses el comando correspondiente lo cual envia desde solo datos hasta graficar la señal de un sensor. La conexión en maquina virtual en veces da problemas con lo que es necesario reiniciarla y de nuevo ejecutar el comando para darle los permisos al puerto para que se comunique con ROS.