



UNIVERSIDAD POLITÉCNICA
DE LA ZONA METROPOLITANA DE GUADALAJARA

Dinámica de Robots.

Chagoya de la Cruz Levi Hazael.

Tarea 7. Ley de Control.

Noveno Cuatrimestre.

UPZMG.

Ingeniería en Mecatrónica.

Mtro. Enrique Morán Garabito.

Objetivos.

- Realizar una comunicación de ROS y un microcontrolador; en este caso utilizamos el microcontrolador freescale.
- Integrar un sensor a la comunicación y obtener los datos medidos en ROS.

Material:

- Computadora. (ROS previamente instalado)
- Cables de conexión.
- Protoboard.
- Sensor LDR.
- Lámpara LED.

Procedimiento:**Inicialización de ROS.**

```
pi@raspberrypi:~$ export ROS_IP=192.168.0.15
pi@raspberrypi:~$ roscore
... logging to /home/pi/.ros/log/fadd46ea-7550-11e8-9f41-b827eba8d49c/roslaunch-raspberrypi-1576.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://192.168.0.15:33627/
ros_comm version 1.14.1

SUMMARY
=====

PARAMETERS
 * /roscdistro: melodic
 * /rosversion: 1.14.1

NODES

auto-starting new master
process[master]: started with pid [1586]
ROS_MASTER_URI=http://192.168.0.15:11311/

setting /run_id to fadd46ea-7550-11e8-9f41-b827eba8d49c
process[rosout-1]: started with pid [1600]
started core service [/rosout]
█
```

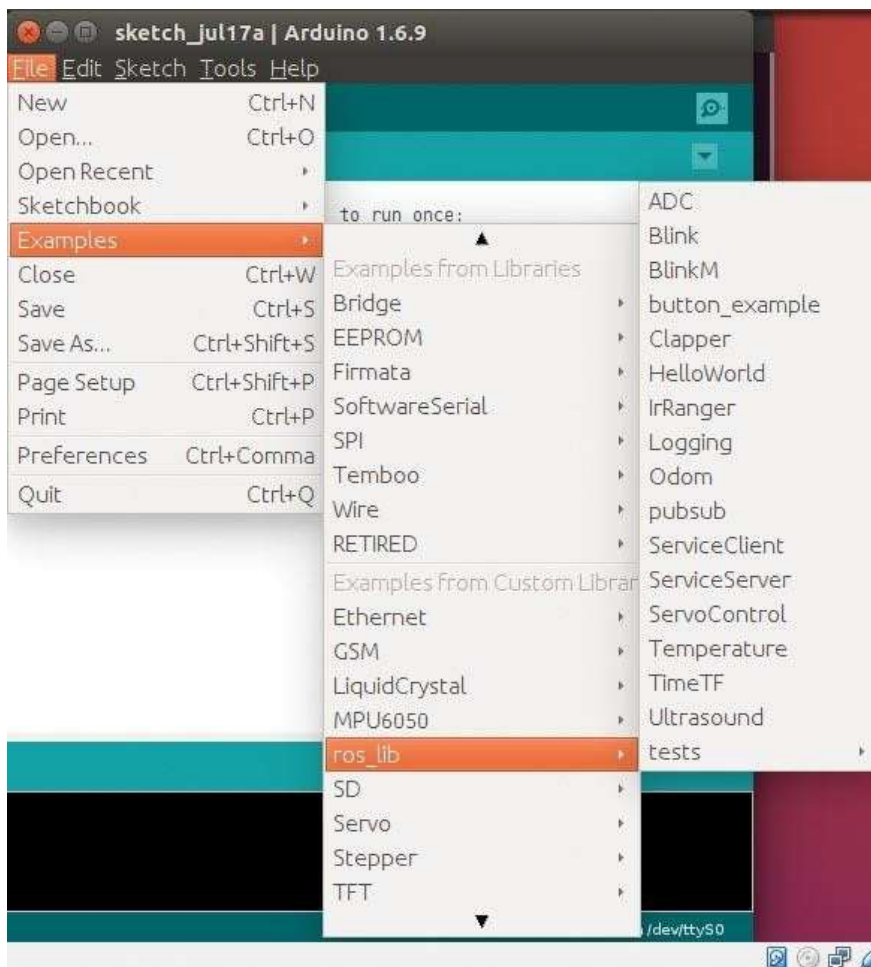
1.- Ingresar esta línea de texto en la terminal para empezar a instalar las librerías de Arduino.

```
sudo apt-get install ros-indigo-rosserial-arduino
```

2.- Crear un catkin en Arduino con los siguientes comandos:

```
git clone https://github.com/ros-drivers/rosterial.git
cd <ws> catkin_make rosrn rosterial_arduino
make_libraries.py . rosrn rosterial_arduino
make_libraries.py .
```

3.- Reiniciar nuestro ordenador y ya deberíamos de tener las librerías de ros en Arduino.



4.- Abrimos el ejemplo de Blink el cual sería el del led y nos abrirá un código como el siguiente:



4.- Arduino en ros usaremos los siguientes comandos:

```
roscore
```

Después ubicamos el puerto y ponemos:

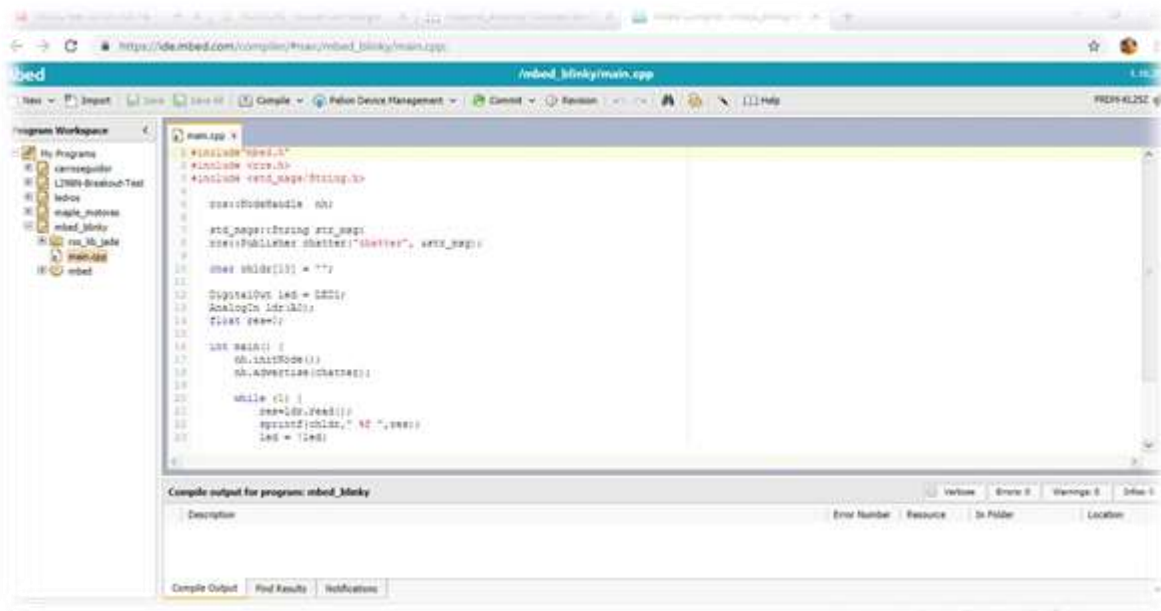
```
roslaunch rosserial_python serial_node.py /dev/ttyUSB0
```

Después:

```
rostopic pub toggle_led std_msgs/Empty --once
```

5.-. Una vez hecho lo de Arduino en mbed importamos la librería de ros y hacemos un código para un sensor. Yo utilice el LDR.

CODIGO:



6.- Ya teniendo el código del sensor en mbed con ros corremos el programa utilizando los mismos comandos que Arduino solo que esta vez imprimiremos caracteres.

```
roscore rosrund rosserial_python serial_node.py
/dev/ttyUSB0 rostopic echo chatter
```