
Robots móviles

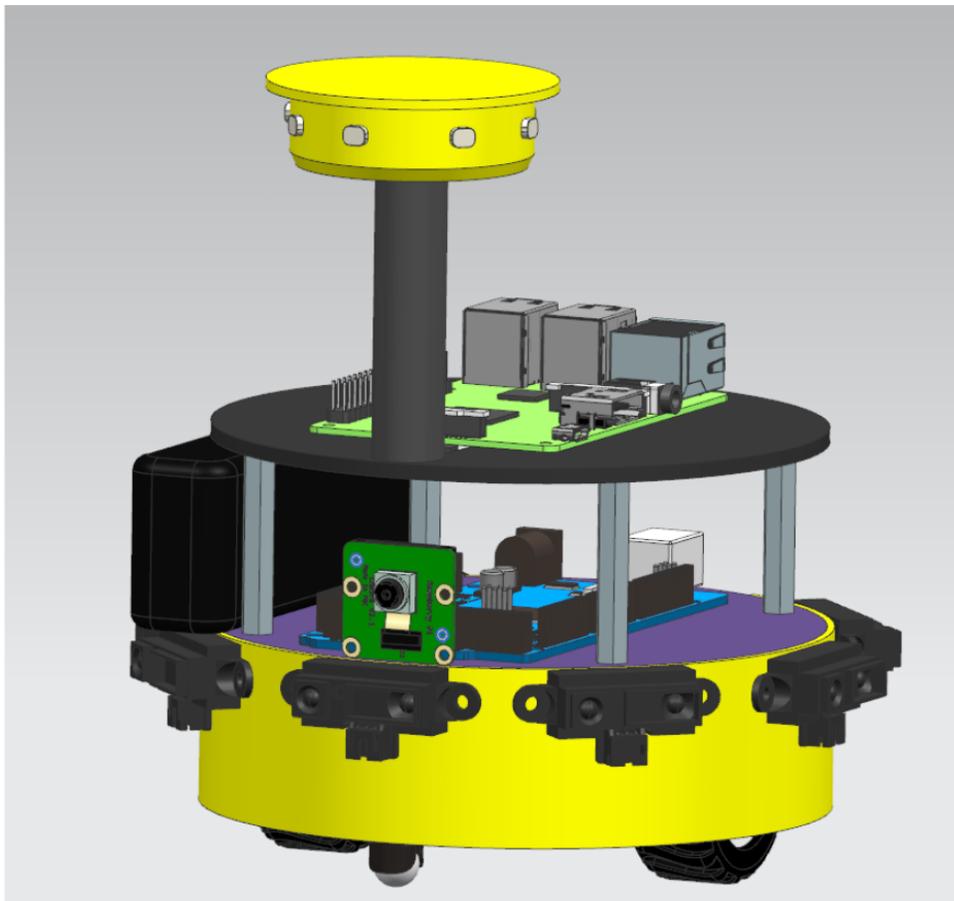
práctica N°5

Programación de Minibot

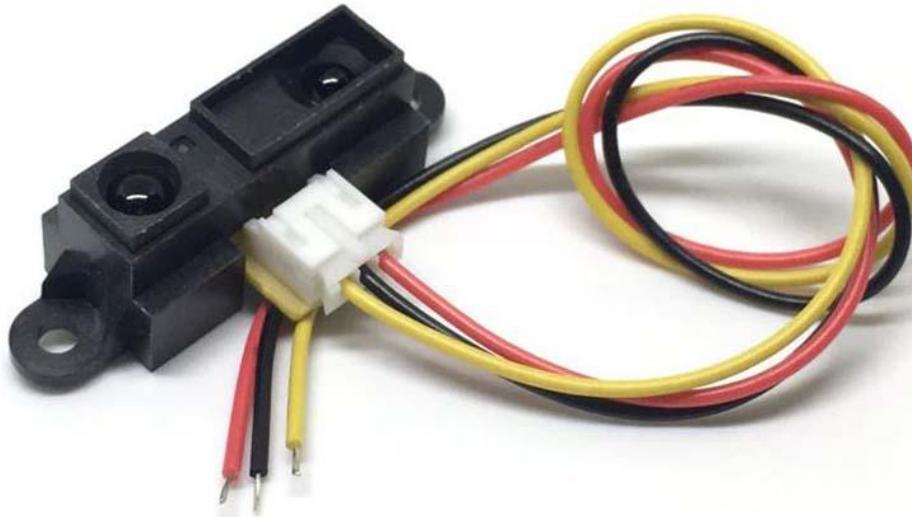
Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE100821

Objetivo: Que el alumno aprenda las diferencias entre el comportamiento de un robot en un ambiente simulado y el de un robot real y usar esta información para disminuir el real gap.

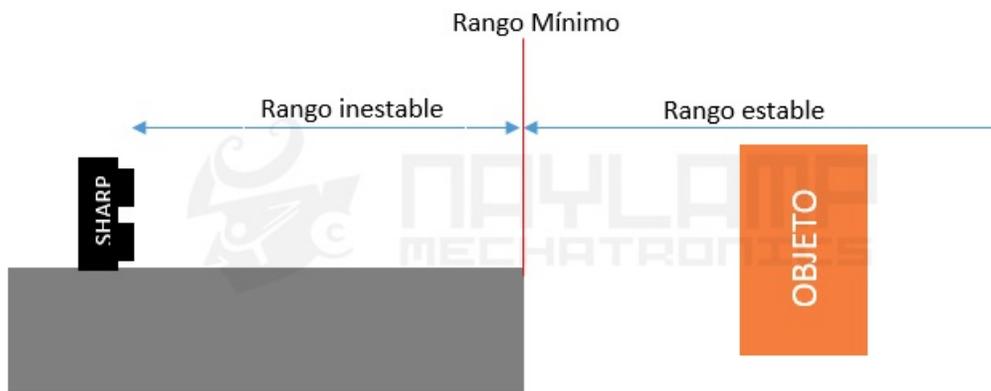
Configuración del Minibot:



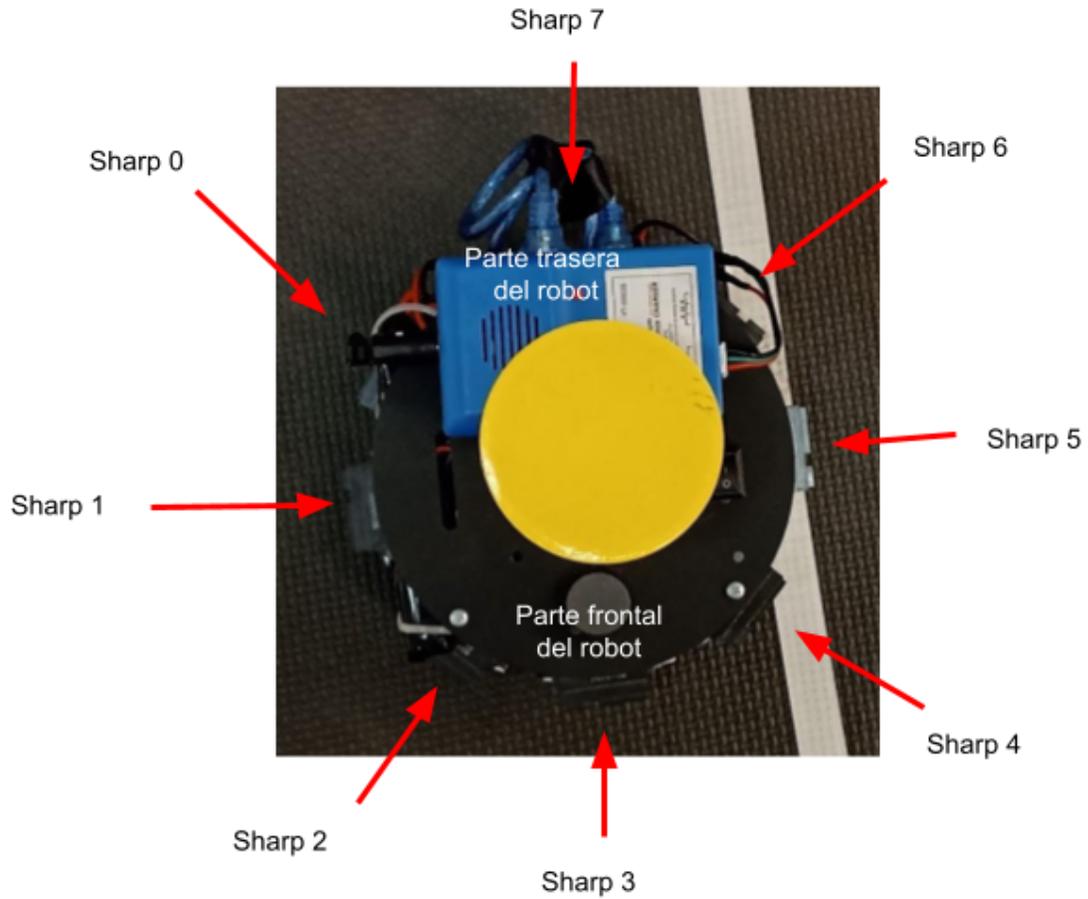
Este robot cuenta con 8 sensores sharp, un sensor sharp es un dispositivo capaz de medir la distancia entre él y un objeto cercano con la ayuda de un sensor infrarrojo y un receptor para así mediante triangulación poder obtener la distancia.



Los sensores sharp utilizados en este robot tienen un rango de medición entre 4 cm y 30 cm por lo que puede que existan lecturas erróneas con objetos que se encuentren muy cerca.



La configuración de los sensores es la siguiente:

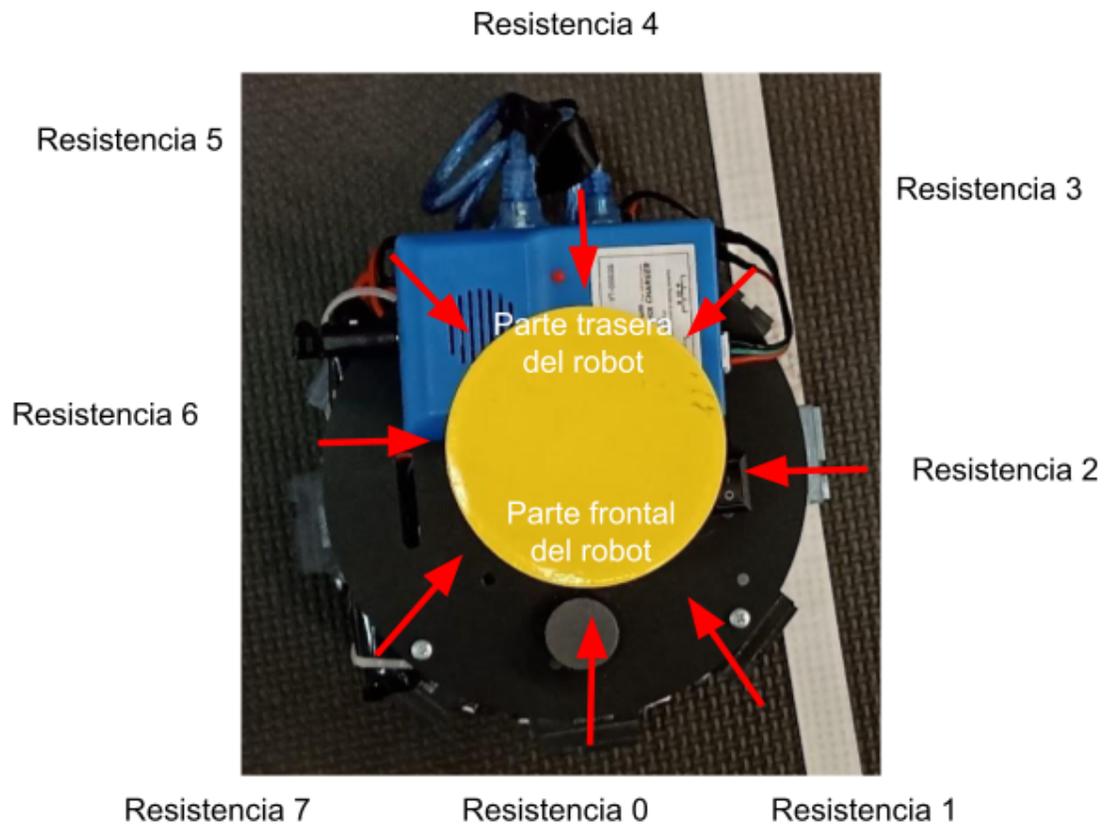


Comienza en el sensor 0 a un costado del robot y en sentido inverso a las manecillas del reloj cada 45° se encuentra el 1 y así sucesivamente hasta llegar al sensor 7 en la parte trasera del robot.

De igual forma el robot tiene 8 sensores de luz o fotoresistencias las cuales permiten medir los niveles de luminosidad del ambiente.



Estos sensores se encuentran distribuidos de la siguiente manera:



Descarga de Any Desk:

Para la realización de la conexión remota debemos descargar el programa Any Desk de la siguiente manera:

Diríjase al siguiente link el cual lo llevará a la página oficial de Any Desk y descargue el paquete para Debian Ubuntu

<https://anydesk.com/es/downloads/linux>

anydesk.com/es/downloads/linux

Dark Mode

Español

AnyDesk

Por qué AnyDesk

Soluciones

Precios

Servicios

Empresa

Prueba Gratuita

Descargas

Descubre AnyDesk para Linux

Su software de escritorio remoto para Linux

- Operación fácil y estable.
- Potente conectividad remota basada en Linux.
- Acceso remoto fluido y sin problemas a cualquier ordenador.
- Conexiones continuas a cualquier sistema operativo.
- Herramientas de administración y configuración sencillas y fáciles de usar.

Use DEB Repository

Use RPM Repository

Debian / Ubuntu / Mint (64 Bit)

openSUSE 15/42 (64 Bit)

CentOS 7 (64 Bit)

CentOS 8 (64 Bit)

RedHat Enterprise Linux 7 (64 Bit)

RedHat Enterprise Linux 8 (64 Bit)

Generic Linux (64 Bit)

Otras versiones

Vedi tutti i sistemi operativi supportati.

https://anydesk.com/es/downloads/thank-you?dv=deb_64

anydesk_6.2....deb

Mostrar

Una vez hecho esto localice el paquete .deb en su carpeta de descargas y dando click derecho seleccione la opción “abrir con instalación de software” y aparecerá la siguiente pantalla.

anydesk

anydesk

The Fastest remote desktop software on the market.

Instalar

It allows for new usage scenarios and applications that have not been possible with current remote desktop software.

Sitio web

Detalles

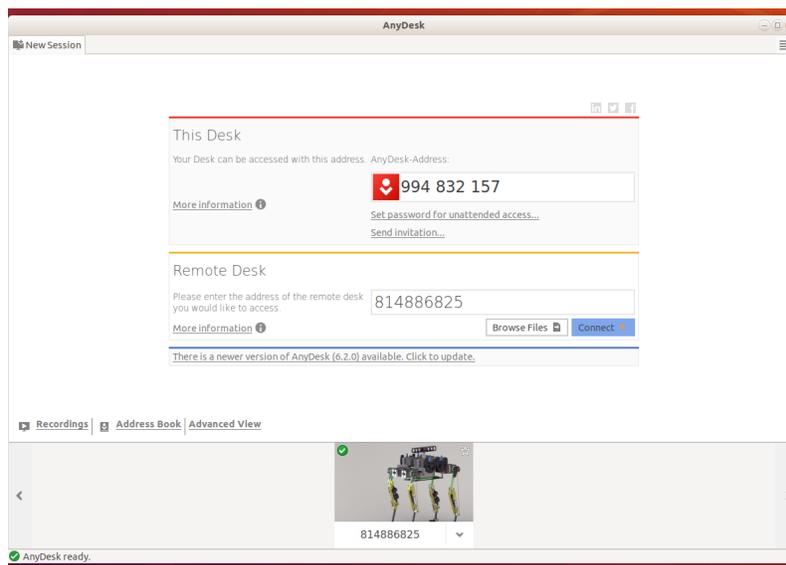
Versión	6.2.0
Actualizada	Nunca
Licencia	Privativa
Origen	anydesk_6.2.0-1_amd64.deb
Tamaño descargado	0 bytes

Para terminar damos click en el botón de instalar para que ésta se inicie y una vez terminada podemos ir a nuestro buscador de archivos para localizar el programa.

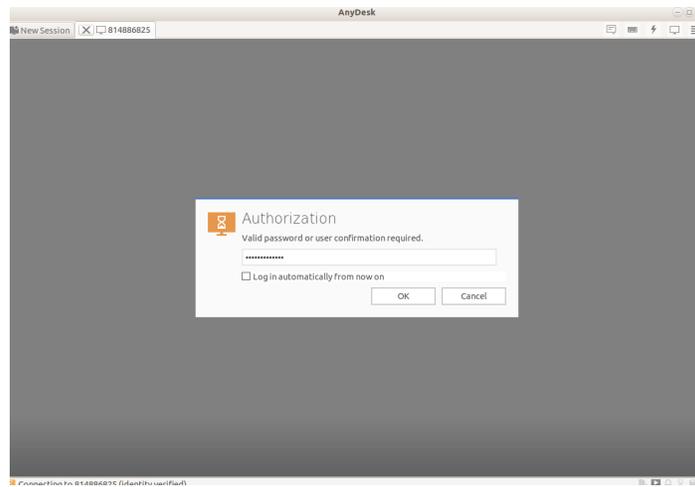


Conexión remota al laboratorio:

Abrimos el programa y escribimos la dirección **814886825** para conectarse al servidor del laboratorio remoto:



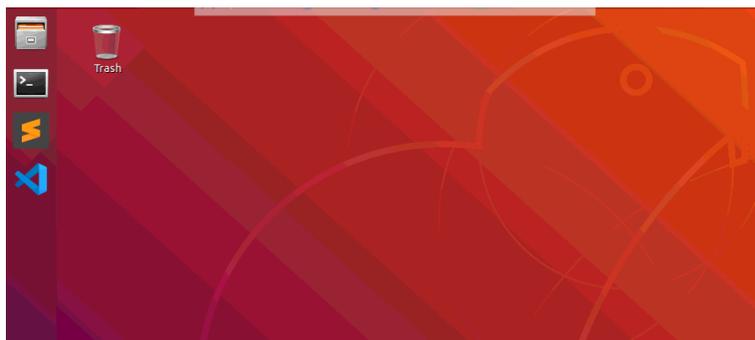
Le pedirá una contraseña para acceder, ingrese con: **pumas-alumnos**



Una vez conectado al escritorio remoto, inicie sesión con el usuario que se le haya asignado:



La contraseña correspondiente es: **pumas-aluX** , donde **X** es el número de alumno de la sesión. Iniciada la sesión se debe mostrar la pantalla principal, tal y como se ve en la siguiente imagen:



Abra una terminal dentro del servidor remoto y conéctese al robot con el alias **minibot_1** o dependiendo el minibot que esté utilizando (1, 2, 3, etc.). La terminal le debe dar el mensaje de bienvenida para el sistema del robot.

```
pi@raspberrypi: ~/Minibot/catkin_ws
File Edit View Search Terminal Help
Welcome to Remote Laboratory System [.][]
alumno_1@remote-lab:~/MobileRobotSimulator/catkin_ws$ minibot_1
Linux raspberrypi 4.9.80-v7+ #1098 SMP Fri Mar 9 19:11:42 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr 25 17:52:59 2022 from 192.168.0.143

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

W3lc0me to MiniBot System []_[]
pi@raspberrypi:~/Minibot/catkin_ws $
```

***Si el sistema no se conecta probablemente el robot no esté encendido.**

Dentro de la terminal de la Raspberry Pi teclee:

roslaunch surge_et_ambula minibot.launch

para levantar el nodo maestro y los principales nodos del hardware del robot.

```
File Edit View Search Terminal Help
pi@raspberrypi:~/Minibot/catkin_ws $ roslaunch surge_et_ambula minibot.launch
... logging to /home/pi/.ros/log/8b03cdfc-c4da-11ec-b1f0-b827eb7b2a71/roslaunch-
raspberrypi-1609.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.

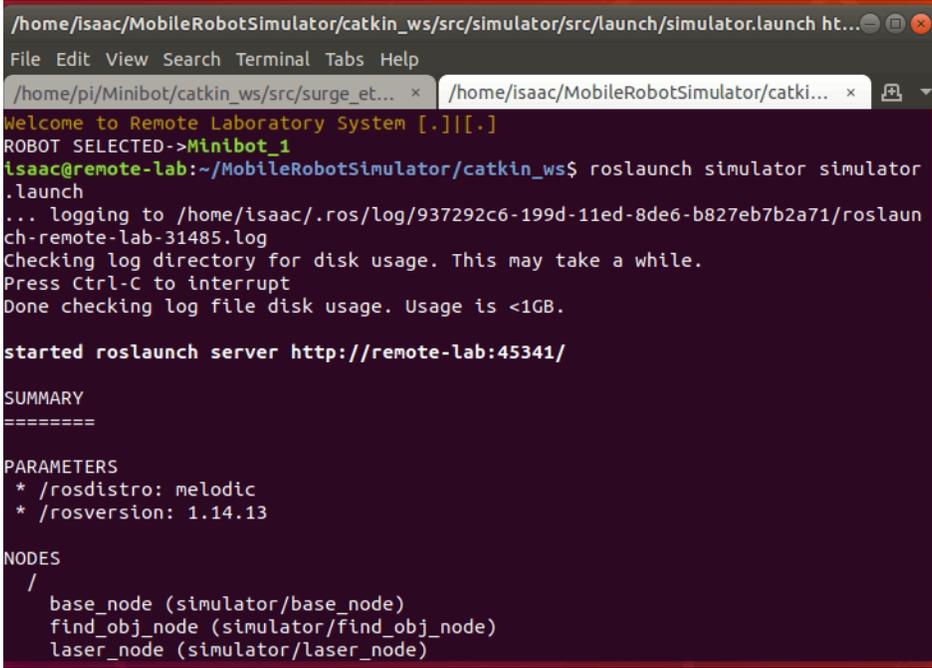
started roslaunch server http://raspberrypi:37551/

SUMMARY
=====

PARAMETERS
* /robot_description: <robot name="carr...
* /rostdistro: kinetic
* /rosterial_python/baud: 200000
* /rosterial_python/port: /dev/ttyUSB0
* /rosversion: 1.12.9

NODES
/
  battery_reader (battery_reader/battery_reader.py)
  camera_node (camera/camera_node)
  laser_scan_node (laser_scan/laser_scan_node)
  mobile_base_node (mobile_base/mobile_base_node)
  robot_state_publisher (robot_state_publisher/state_publisher)
  rosterial_python (rosterial_python/serial_node.py)
```

Abra otra terminal y teclee el siguiente comando para levantar el simulador: **roslaunch simulator simulator.launch**



```
/home/isaac/MobileRobotSimulator/catkin_ws/src/simulator/src/launch/simulator.launch ht...
File Edit View Search Terminal Tabs Help
/home/pi/Minibot/catkin_ws/src/surge_et... x /home/isaac/MobileRobotSimulator/catki... x
Welcome to Remote Laboratory System [.][]
ROBOT SELECTED->Minibot_1
isaac@remote-lab:~/MobileRobotSimulator/catkin_ws$ roslaunch simulator simulator
.launch
... logging to /home/isaac/.ros/log/937292c6-199d-11ed-8de6-b827eb7b2a71/roslau
ch-remote-lab-31485.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://remote-lab:45341/

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.13

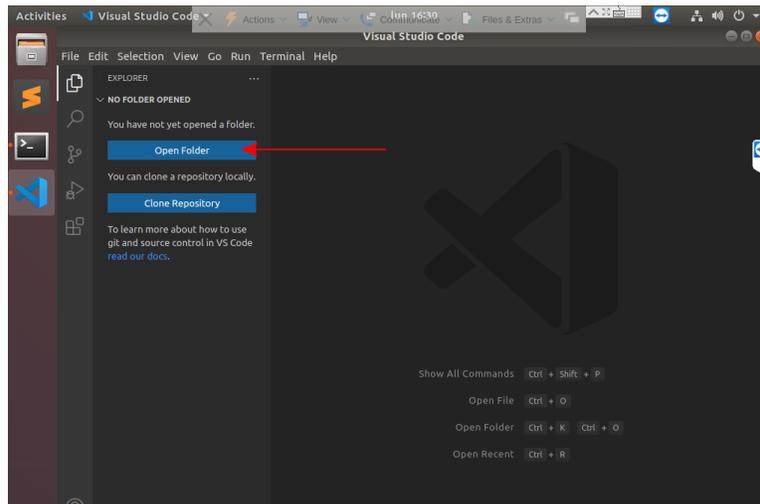
NODES
/
  base_node (simulator/base_node)
  find_obj_node (simulator/find_obj_node)
  laser_node (simulator/laser_node)
```

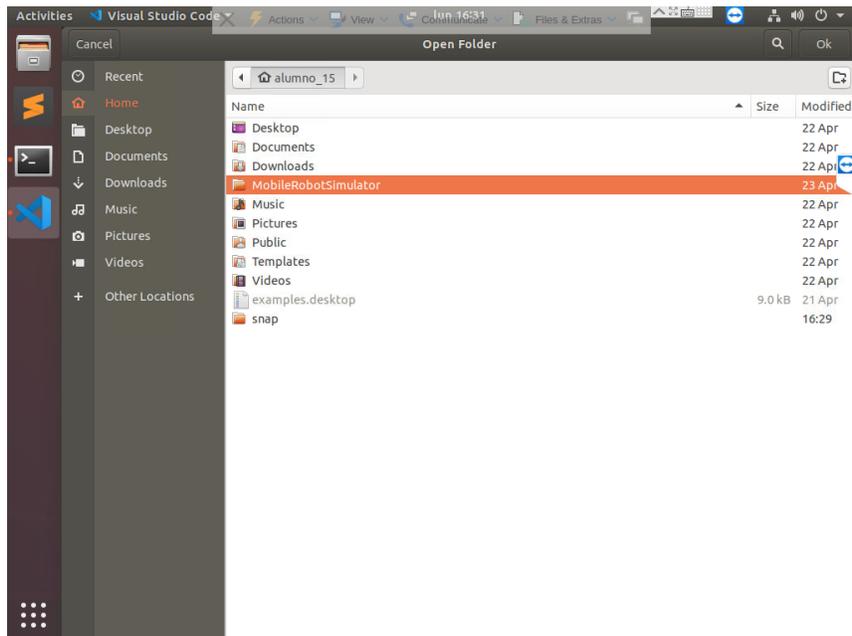
Una vez abierto el simulador puede comenzar a hacer pruebas de los algoritmos ya cargados en el sistema o usted mismo subir sus propios algoritmos.

Cargar algoritmos:

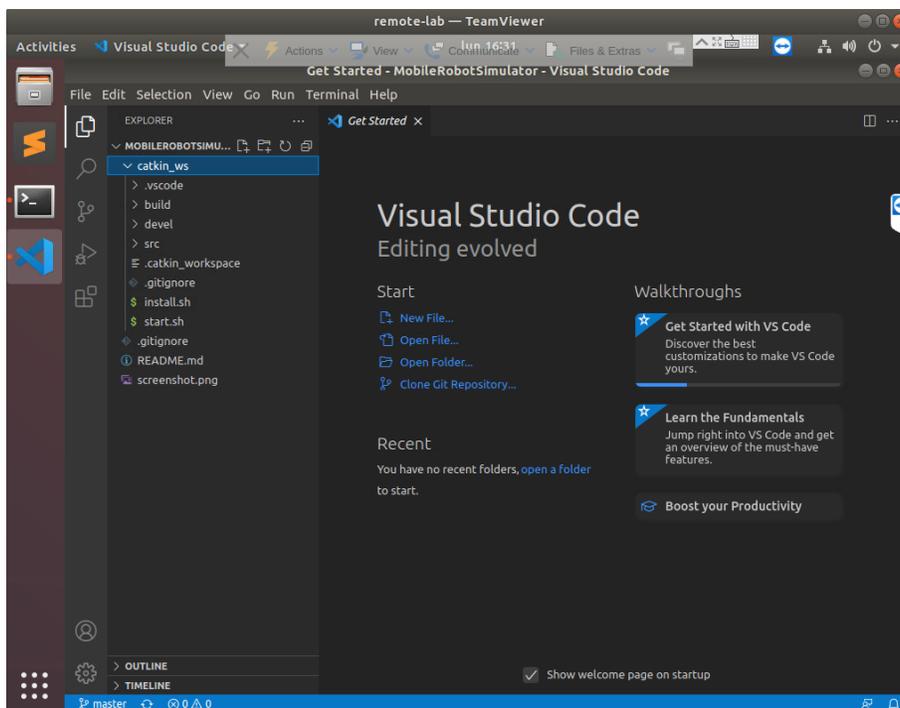
Tiene dos opciones para poder modificar el código del robot, puede trabajar en la máquina virtual o hacer las pruebas directamente en su computadora para después enviar su código.

Para modificar el código en la máquina virtual, utilice el editor de su conveniencia, en este caso VSC. Cliqueamos Open Folder para cargar todo el repositorio.





Dentro de la carpeta se deben visualizar todos los archivos del repositorio



En el directorio **catkin_ws/src/simulator/src/state_machines** es donde se implementan los comportamientos que se ven en la GUI.

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the directory structure of the `catkin_ws` workspace. The file `user_sm.h` is selected in the `state_machines` directory. The main editor displays the content of `user_sm.h`, which includes a header section with a license notice and a state machine function definition.

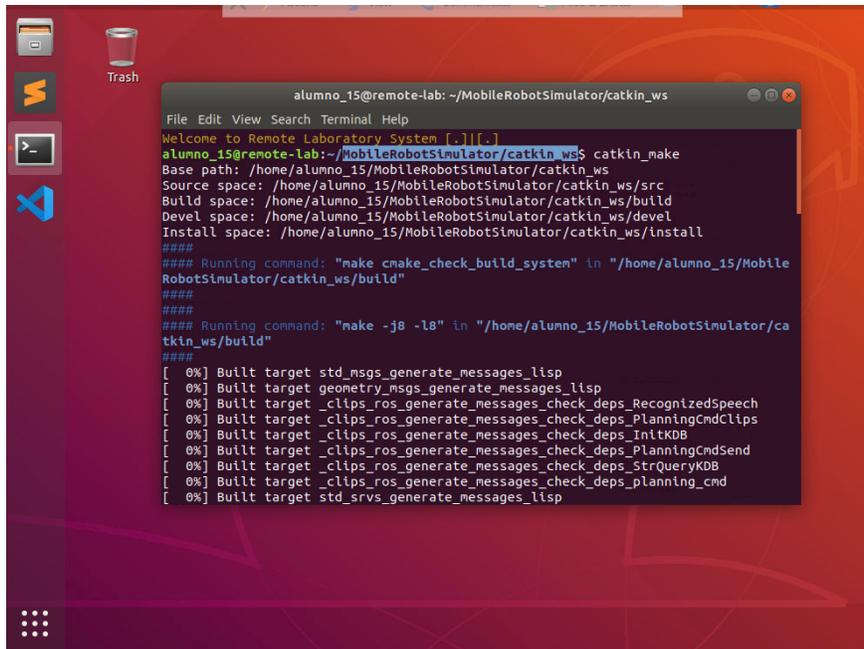
```
1 //*****  
2 // user_sm.h  
3 //*****  
4  
5 user_sm.h  
6  
7 FI-UNAM  
8 17-2-2019  
9 //*****  
10  
11  
12  
13  
14 // State Machine  
15 void user_sm(float intensity, float *light_values, float *obs  
16             movement *movements ,int *next_state, fl  
17 {  
18  
19     int state = *next_state;  
20     int i;  
21  
22     printf("intensity %f\n",intensity);  
23     printf("quantized destination %d\n",dest);  
24     printf("quantized obs %d\n",obs);  
25  
26     for(int i = 0; i < 8; i++)  
27         printf("light_values[%d] %f\n",i,light_values[i]);  
28     for (int i = 0; i < size; i++)  
29         printf("laser observations[%d] %f\n",i,observations[
```

`user_sm` es el archivo disponible para que los alumnos implementen y prueben sus algoritmos.

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the directory structure of the `MobileRobotSimulator` workspace. The file `user_sm.h` is selected in the `state_machines` directory. The main editor displays the content of `user_sm.h`, which is identical to the one shown in the previous screenshot.

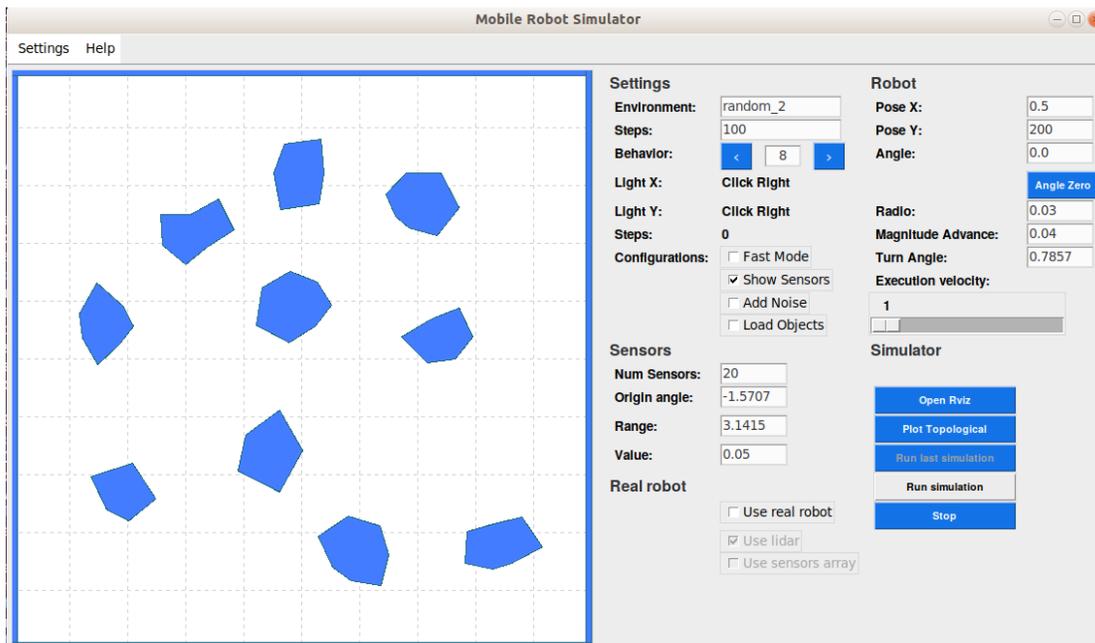
```
1 //*****  
2 // user_sm.h  
3 //*****  
4  
5 user_sm.h  
6  
7 FI-UNAM  
8 17-2-2019  
9 //*****  
10  
11  
12  
13  
14 // State Machine  
15 void user_sm(float intensity, float *light_values, float *obs  
16             movement *movements  
17 {  
18  
19     int state = *next_state;  
20     int i;  
21  
22     printf("intensity %f\n",intensity);  
23     printf("quantized destination %d\n",dest);  
24     printf("quantized obs %d\n",obs);  
25  
26     for(int i = 0; i < 8; i++)  
27         printf("light_values[%d] %f\n",i,light_values[i]);  
28     for (int i = 0; i < size; i++)  
29         printf("laser observations[%d] %f\n",i,observations[
```

Cada que haga modificaciones en el archivo, compile con `catkin_make` dentro de una terminal. No olvide que debe estar dentro de las carpeta: **MobileRobotSimulator/catkin_ws**

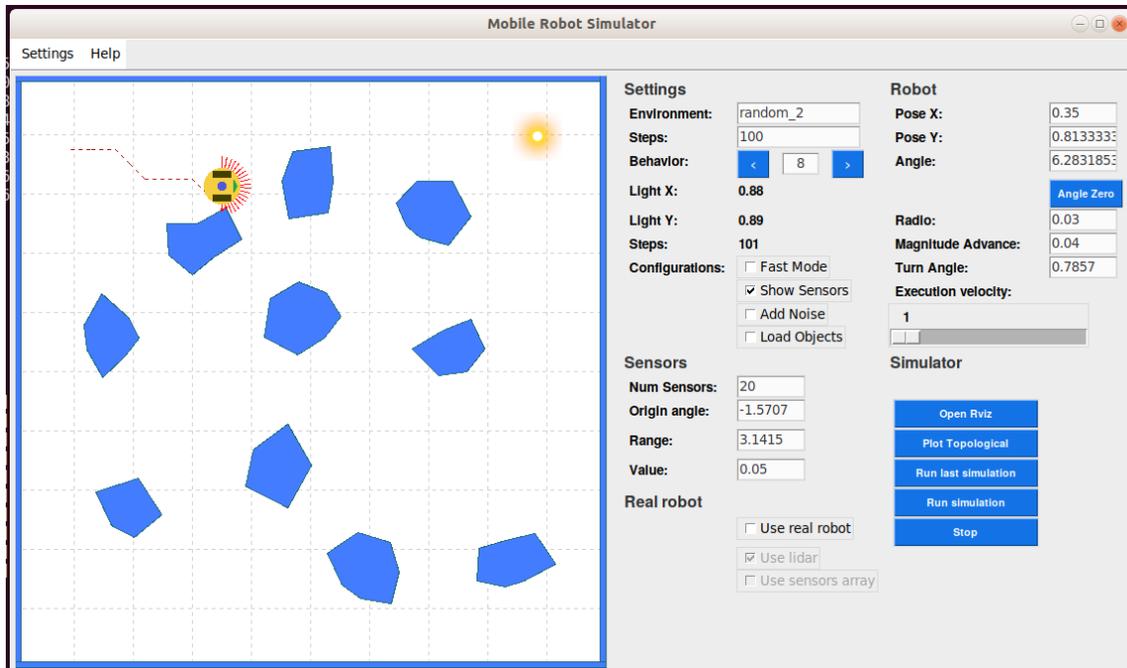


```
alumno_15@remote-lab: ~/MobileRobotSimulator/catkin_ws
File Edit View Search Terminal Help
Welcome to Remote Laboratory System [...][...]
alumno_15@remote-lab:~/MobileRobotSimulator/catkin_ws$ catkin_make
Base path: /home/alumno_15/MobileRobotSimulator/catkin_ws
Source space: /home/alumno_15/MobileRobotSimulator/catkin_ws/src
Build space: /home/alumno_15/MobileRobotSimulator/catkin_ws/build
Devel space: /home/alumno_15/MobileRobotSimulator/catkin_ws/devel
Install space: /home/alumno_15/MobileRobotSimulator/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/alumno_15/Mobile
RobotSimulator/catkin_ws/build"
####
#### Running command: "make -j8 -l8" in "/home/alumno_15/MobileRobotSimulator/ca
tkin_ws/build"
####
[ 0%] Built target std_msgs_generate_messages_lisp
[ 0%] Built target geometry_msgs_generate_messages_lisp
[ 0%] Built target _clips_ros_generate_messages_check_deps_RecognizedSpeech
[ 0%] Built target _clips_ros_generate_messages_check_deps_PlanningCmdClips
[ 0%] Built target _clips_ros_generate_messages_check_deps_InitKDB
[ 0%] Built target _clips_ros_generate_messages_check_deps_PlanningCmdSend
[ 0%] Built target _clips_ros_generate_messages_check_deps_StrQueryKDB
[ 0%] Built target _clips_ros_generate_messages_check_deps_planning_cmd
[ 0%] Built target std_srvs_generate_messages_lisp
```

Una vez compilado su archivo puede ir al simulador seleccionar el comportamiento número 8 y presionar Run simulation para ejecutar la simulación de su algoritmo.

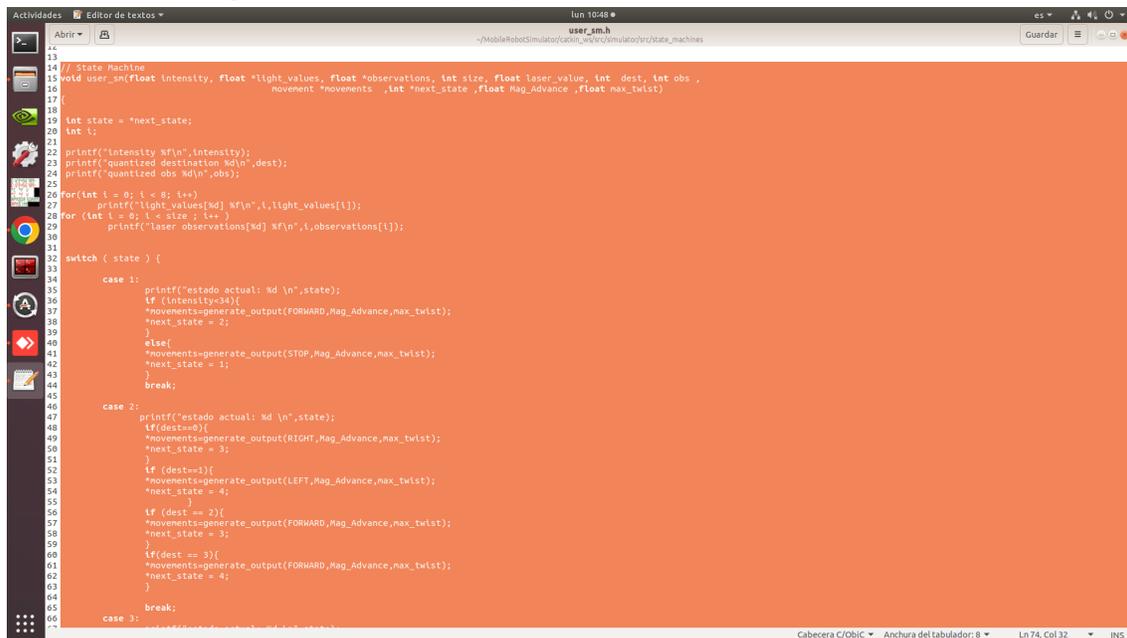


Para posicionar al robot necesita dar click izquierdo sobre el mapa en el punto donde desee que inicie el robot y luego click derecho para marcar la fuente luminosa que será el punto al que deba llegar, automáticamente comenzará la simulación después de posicionar el objetivo.

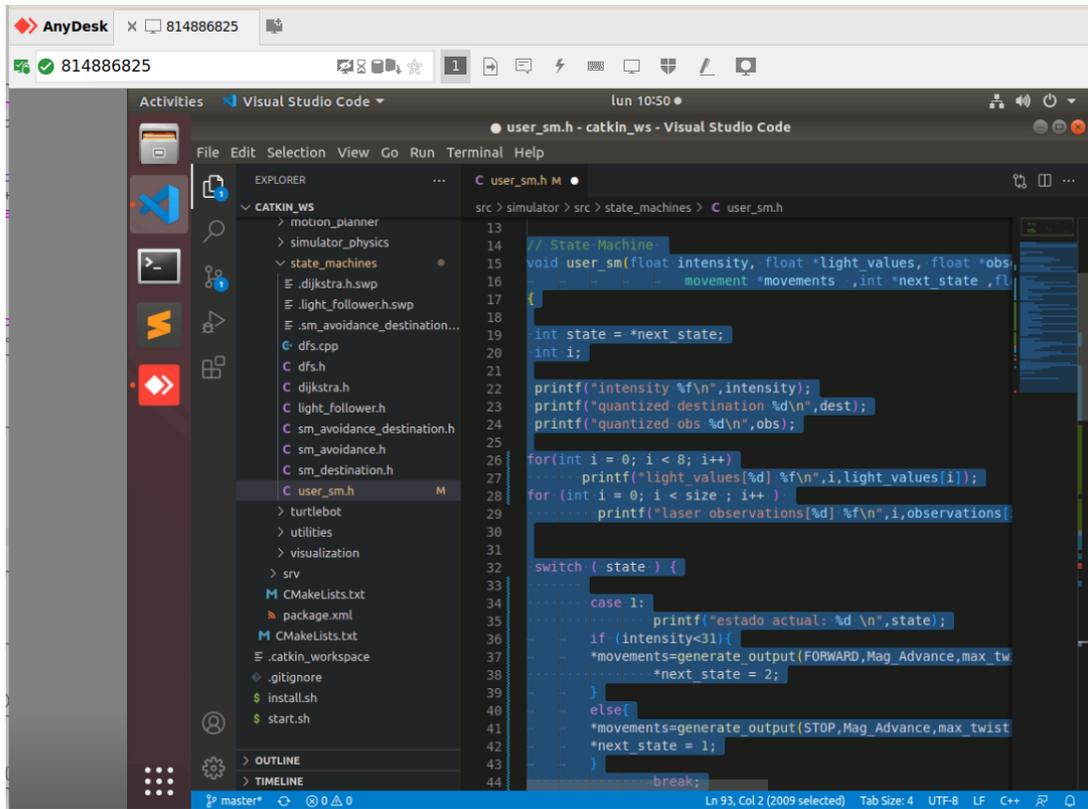


Esto puede realizarlo de igual forma en su computadora y una vez que haya realizado las pruebas necesarias simplemente puede copiar el código desde su computadora y pegarlo en el editor de Any Desk, compilarlo y ejecutarlo.

seleccione su código, dé click derecho, copiar



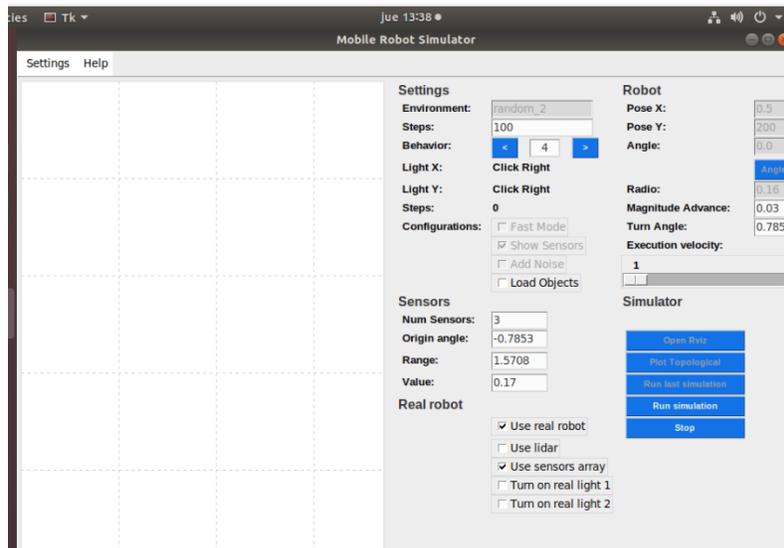
Vaya al editor en Any Desk, click derecho y pegar, para ejecutarlo necesita compilarlo y realizar los pasos descritos anteriormente.



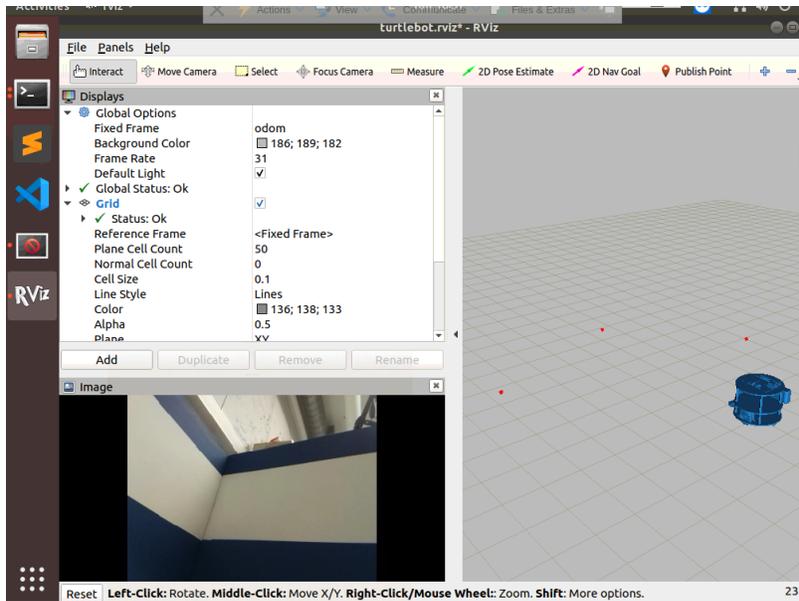
Usar el robot real:

Una vez terminadas sus pruebas en el simulador ahora puede hacer las pruebas en el robot real del laboratorio para esto dentro de la GUI del simulador de Any Desk clickee las casillas de **Use real robot** y **Use sensors array** para visualizar al robot real dentro de **Rviz**.

[OBJ]



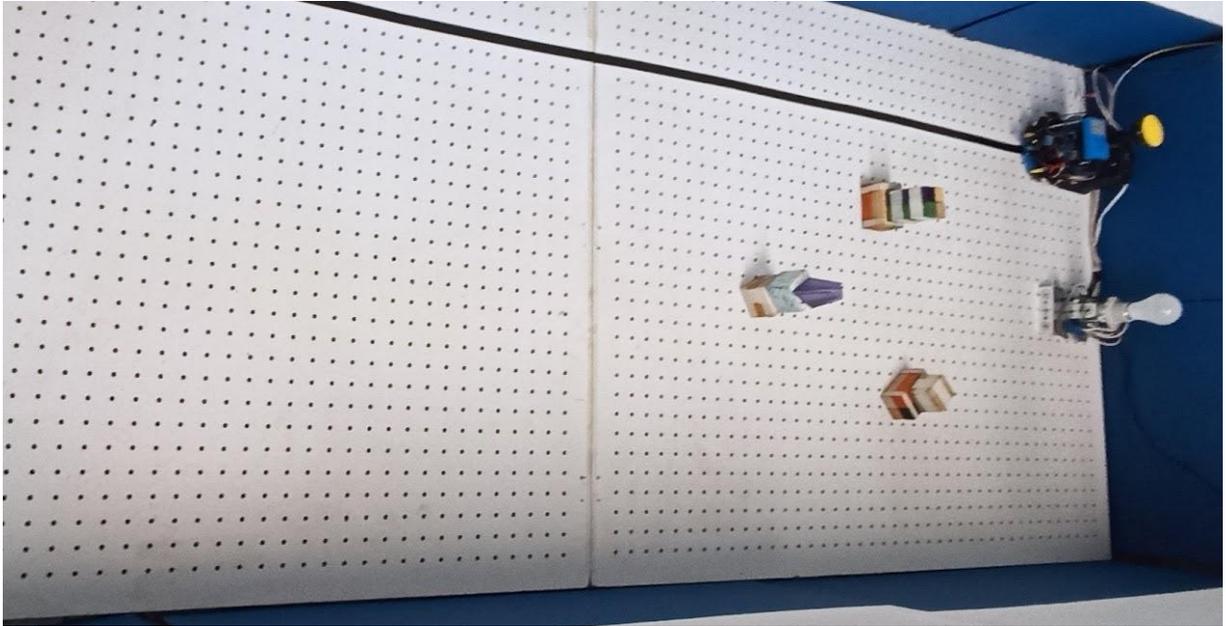
Esto abrirá automáticamente RVIZ, dentro del visualizador se debe poder observar al minibot con la medición de sus sensores y la imagen de su cámara(en algunas ocasiones la imagen de la cámara no está disponible sin embargo esto no afecta al funcionamiento del robot, siempre y cuando pueda ver al robot con las mediciones de sus sensores).



En la GUI tendrá 2 casillas para encender la luces que hay dentro de la arena donde se encuentran los robots.

Vaya al siguiente link para ver la transmisión en vivo del laboratorio virtual:
<https://www.youtube.com/channel/UCyFTPuXtcBiARIOvoWLaLFw>
(La transmisión puede tener un delay de hasta 40 segundos dependiendo de su conexión)

Si el robot se encuentra en su base de carga deberá teleoperarlo para sacarlo de la base:



Para teleoperarlo deberá abrir una nueva terminal y ejecutar el comando:

roslaunch teleop_twist_keyboard teleop_twist_keyboard.py

```
Isaac@remote-lab: ~/MobileRobotSimulator/catkin_ws
File Edit View Search Terminal Tabs Help
/home/pi/Minibot/catkin... x /home/isaac/MobileRob... x isaac@remote-lab: ~/Mo... x
Isaac@remote-lab:~/MobileRobotSimulator/catkin_ws$ roslaunch teleop_twist_keyboard
teleop_twist_keyboard.py
the rosdep view is empty: call 'sudo rosdep init' and 'rosdep update'

Reading from the keyboard and Publishing to Twist!
-----
Moving around:
  u   i   o
  j   k   l
  m   ,   .

For Holonomic mode (strafing), hold down the shift key:
-----
  U   I   O
  J   K   L
  M   <   >

t : up (+z)
b : down (-z)

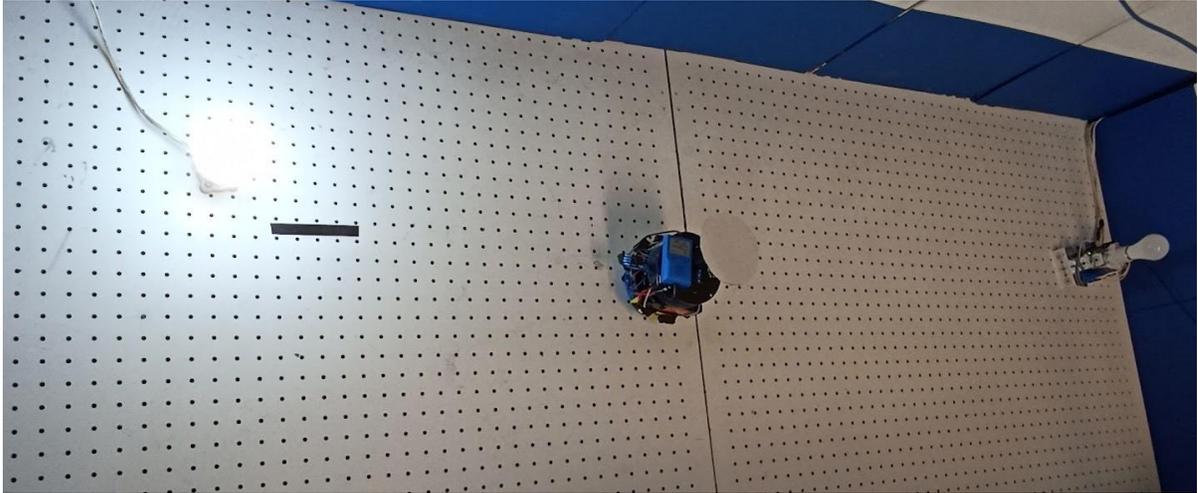
anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
```

Ahora puede mover el robot utilizando sus teclas **i,j,l,k** y **m**, con la tecla **i** el robot se moverá hacia enfrente, con **j** gira hacia la izquierda, con **l** gira a la derecha, con **m** va hacia atrás y con **k** el robot se detiene.

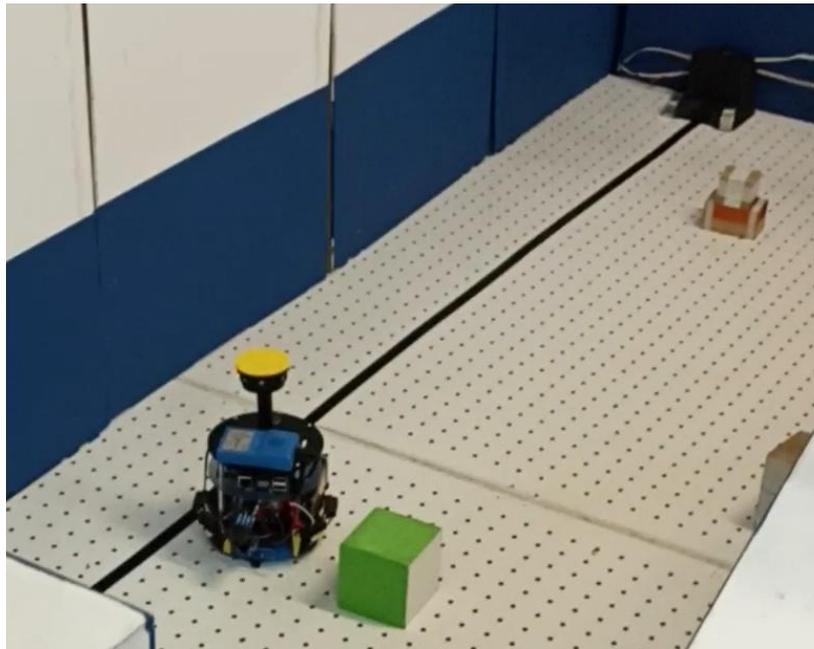
cada que presione una tecla el robot se moverá hasta que presione la tecla **k** y lo detenga por lo que se recomienda ver mediante la transmisión en vivo cada movimiento que haga para evitar que choque con algo o se quede atorado, de igual forma al principio es recomendable presionar una tecla de dirección y enseguida la tecla **k** para detenerlo, esto para realizar movimientos controlados(deberá tener en cuenta el delay en la transmisión).

Una vez fuera de la base deberá encender el foco más alejado del robot marcando las casillas “turn on real light 1” o “turn on real light 2” dependiendo de cual sea la más alejada y ejecutar el comportamiento 4 para hacer pruebas y ver como el robot evita obstáculos.



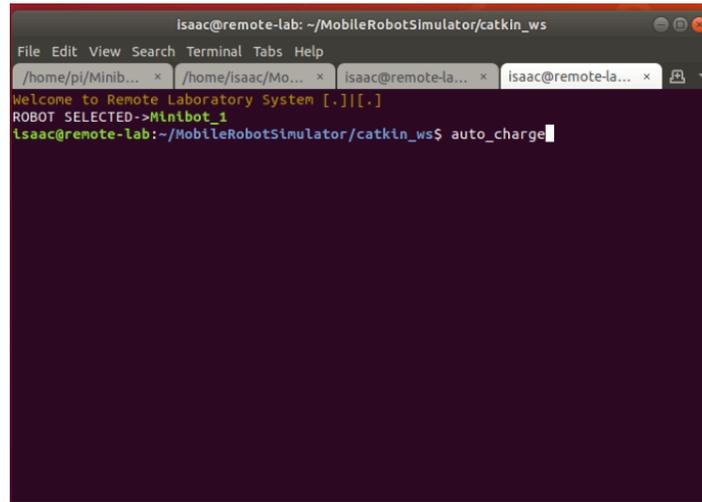
Para terminar puede ahora seleccionar el comportamiento 8 y presionar Run simulation para ejecutar su código y ver mediante las cámaras lo que su algoritmo hace.

Una vez terminadas sus pruebas deberá llevar al robot nuevamente a su centro de carga, para eso se diseñó un servicio de ROS seguidor de línea el cual permitirá al usuario llevar de forma sencilla al robot hacia su posición inicial, para llamar a este servicio primero deberá teleoperar al robot y llevarlo lo más cerca que pueda de la línea tratando de alinearlos de forma paralela a ésta y orientandolo en dirección a su base de carga(No importa que el robot no esté exactamente sobre la línea) de la siguiente manera.



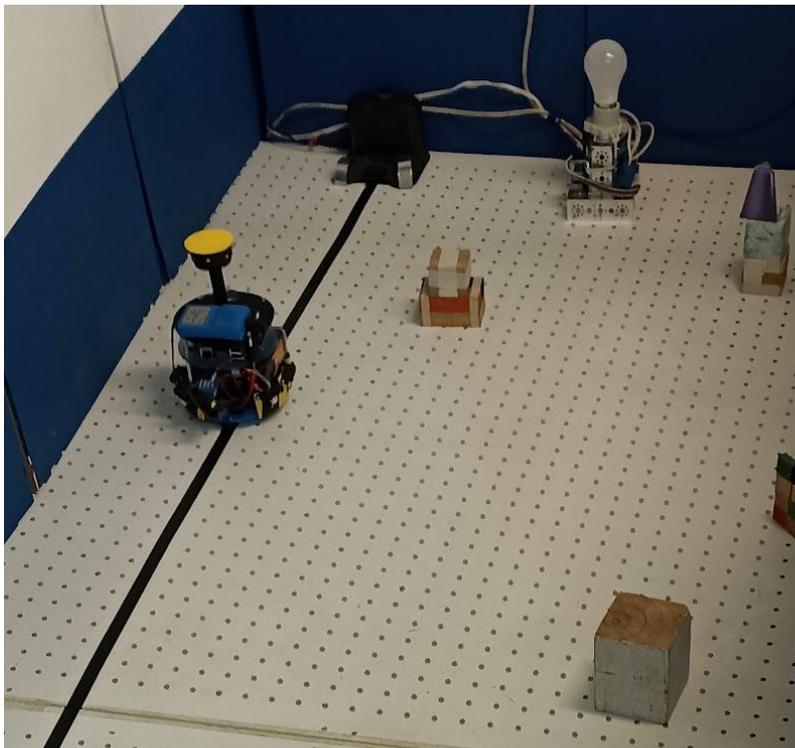
Una vez que el robot esté en dirección al cargador deberá abrir una nueva terminal y escribir en ella el comando:

auto_charge



```
isaac@remote-lab: ~/MobileRobotSimulator/catkin_ws
File Edit View Search Terminal Tabs Help
/home/pi/Minib... x /home/isaac/Mo... x isaac@remote-la... x isaac@remote-la... x
Welcome to Remote Laboratory System [.][]
ROBOT SELECTED->Mintbot_1
isaac@remote-lab:~/MobileRobotSimulator/catkin_ws$ auto_charge
```

Automáticamente el robot comenzará a avanzar para buscar la línea, una vez que la encuentre se posicionará sobre ella y la seguirá hasta el final, cuando el servicio detecte que el robot se está cargando se detendrá por sí solo.



En este punto ya puede cerrar todas terminales abiertas y dar click en Switch user para cerrar su sesión.

