

# Tarea Modulo 03

Profesor: Jorge Díaz  
jjdiaz6@uc.cl

Ayudante: Alonso Rivera  
adriviera1@uc.cl

## 1. Parte practica

Para la realizacion de esta tarea es necesaria la utilizacion del programa [Webots](#), compatible con Windows, Linux y macOS ([Guia de instalacion](#)), tambien es necesario contar con Python 3 (preferentemente [Python3.12](#)) y se recomienda fuertemente el uso de la libreria [Numpy](#) para agilizar los calculos.

### 1.1. Simulacion

En este caso utilizaremos un robot modelo “Turtlebot Burguer” con las siguientes características:

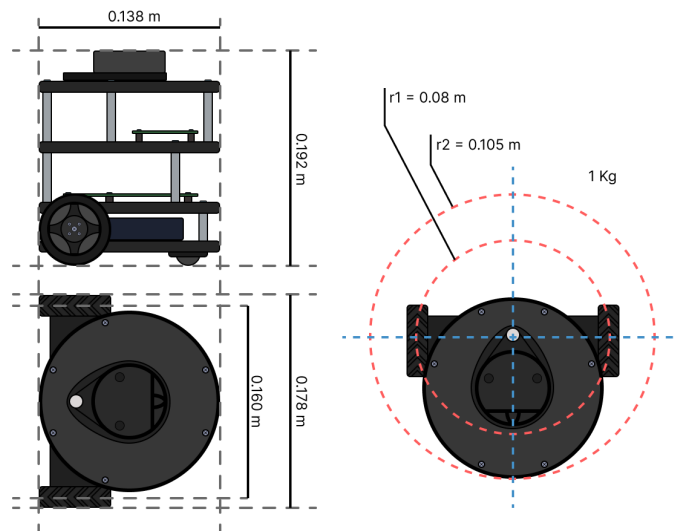


Figure 1: Turtlebot burguer

HINT: el robot cuenta con un modulo “[gps](#)” y un modulo “[gyro](#)” que junto con la funcion integrada en Python “[dir\(\)](#)” te seran utiles.

Puedes descargar la estructura para la tarea clonando [este repositorio](#).

#### 1.1.1. Mapeo

Para esta parte deberas hacer uso del modulo LiDAR LDS-01 integrado en tu Turtlebot:

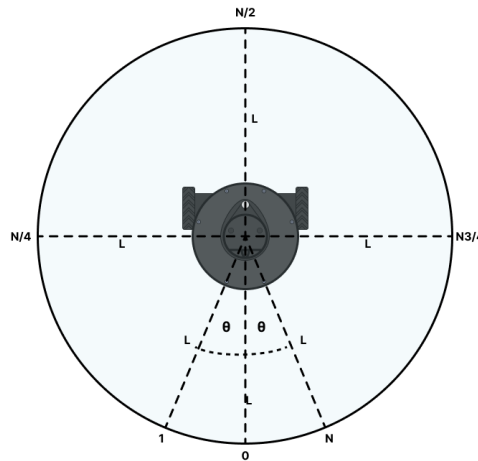


Figure 2: LDS-01

Este toma  $N = 240$  muestras de manera radial, una cada  $\theta = 360/N$  grados, con una distancia maxima de  $L = 1.8$  m de radio y ruido artificial con un coeficiente de 0.0086.

En la carpeta REDACTED debes implementar un controlador que realice lo siguiente:

- Obtenga los datos obtenidos por el sensor lidar
- Registre en un archivo .png el gradiente obtenida por las mediciones
- Dado un movimiento del robot, genere el mapa extendido
- De manera autonoma recorra y mapee todo el laberinto

### 1.1.2. Localizacion

En la carpeta REDACTED debes implementar un controlador que sea capaz de encontrar la ubicacion del robot en cualquier parte del mapa, en este caso se te entrega un codigo donde la funcion gps esta modificada para que solo puedas utilizarlo para detectar el movimiento del robot y no obtener la posicion exacta, tambien se te entrega un mapa en formato .png por lo que puedes decidir usar este o el que generaste en la pregunta anterior.

- Obtener la posicion en el mapa con una precision de un x%
- Obtener la posicion en el mapa con obstaculos que interfieren en las mediciones con una precision de un y%

### 1.1.3. Planificación de Trayectorias

En la carpeta REDACTED debes implementar un codigo que

- Dado un punto x encuentre el camino mas corto a un punto y, es posible realizarlo con el mapa entregado o con el propio realizado en la parte de mapeo
- Bonus ? encontrar el camino mas rapido?, considerando los pesos de perdida al momento de doblar
- Dado un punto x encuentre el camino mas corto a un punto y sin utilizar el modelo interno del mapa, solo utilizando el lidar como sensor de proximidad ( $A^*$ )
- Dado un punto x encontrar el camino mas corto a un punto y, en caso de detectar un objeto en particular, esquivarlo (esta informacion no esta en el mapa, solo es posible saberlo con la camara integrada en el robot dado que los obstaculos no estaran en el rango del lidar)
- Si es que en el camino encuentra otro objeto en particular, resetear la meta

## 1.2. Visión por Computador

- Entrenar un modelo de yolo con un objeto a eleccion, utilizar el modelo y

## 2. Parte teorica

## **2.1. Mapeo**

- Investiga sobre el funcionamiento y aplicacion de los sensores LiDAR en la vida real
- Estos metodos sirven para mas casos que mapas 2d

## **2.2. Localizacion**

## **2.3. Planificación de Trayectorias**

## **2.4. Visión por Computador**