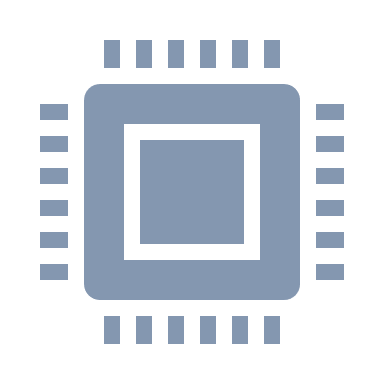
Navigation System

MANUAL TÉCNICO



ÍNDICE

[RESUMEN iv](#_Toc95661194)

[PRESENTACIÓN iv](#_Toc95661195)

[OBJETIVO iv](#_Toc95661196)

[1. Aspectos Teóricos 1](#_Toc95661197)

[1.1 SLAM 1](#_Toc95661198)

[1.2 Odometría 1](#_Toc95661199)

[1.3 Locomoción 1](#_Toc95661200)

[1.5 Holonomía 1](#_Toc95661201)

[1.5 DDMR 2](#_Toc95661202)

[1.6 Sensor Activo 2](#_Toc95661203)

[1.7 ROS 2](#_Toc95661204)

[2. Etapas de funcionalidad del Sistema. 4](#_Toc95661205)

[2.1 Locomoción 4](#_Toc95661206)

[2.2 Mapeo 4](#_Toc95661207)

[2.3 Planificación de caminos y Navegación Autónoma 5](#_Toc95661208)

[3. Aspectos del sistema 7](#_Toc95661209)

[3.1 Hardware 7](#_Toc95661210)

[3.2 Software 7](#_Toc95661211)

[4. Preparación del Sistema 9](#_Toc95661212)

[1. Instalación de Raspbian 9](#_Toc95661213)

[4.2 Configuración de los Buses i2c de la tarjeta Raspberry 13](#_Toc95661214)

[4.3 Instalación de ROS 16](#_Toc95661215)

[4.4nstalación de dependencias necesarias para los módulos de ROS 19](#_Toc95661216)

[4.5 Descarga de los repositorios 21](#_Toc95661217)

[4.5 Instalación del entorno Arduino para Raspberry pi4 22](#_Toc95661218)

[4.6 Instalar librerías para Arduino. 24](#_Toc95661219)

[5. Aspectos de diseño del Sistema de Locomoción y de Odometría 26](#_Toc95661220)

[5.1 Sistema de Locomoción 26](#_Toc95661221)

[5.2 Sistema de Odometría 30](#_Toc95661222)

[6. Esquema de Conexiones 33](#_Toc95661223)

[6.1 Conexión para establecer comunicación I2C 33](#_Toc95661224)

[6.2 Conexión Eléctrica para el Sistema de Locomoción 34](#_Toc95661225)

[6.3 Conexión Eléctrica para el Sistema de Conteo 35](#_Toc95661226)

[6.4 Diagrama de conexiones para el Sistema de Navegación 36](#_Toc95661227)

[7. Códigos y programas 37](#_Toc95661228)

[8. Corrimiento del Sistema 38](#_Toc95661229)

# RESUMEN

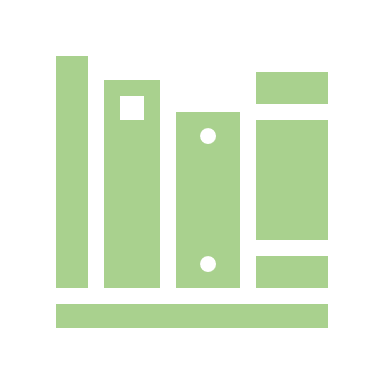
El presente manual detalla los aspectos técnicos e informáticos, mecánicos, eléctricos y electrónicos del Sistema de Navegación. Donde se expone la instalación del SO en el sistema embebido, la instalación de los Frameworks de desarrollo, las dependencias y repositorios necesarios, las variables de diseño para la locomoción y Odometría, la recepción de señales eléctricas e interpretación de estas. Finalmente, la ejecución del Sistema de navegación y del cómo cada una de las partes se retroalimentan.

# PRESENTACIÓN

A vista de dar una detallada explicación del sistema de navegación del proyecto, MEDIC ROBOT, en sus etapas de Locomoción, Mapeo y Navegación autónoma. Dando una explicación acerca del diseño de cada sistema, impresión 3d, esquemas eléctricos y la programación del mismo proyecto, todo con la finalidad de probar, dar mantenimiento, seguimiento y mejoras al sistema de navegación.

# OBJETIVO

Instruir a un futuro colaborador en cada una de las etapas del Sistema de Navegación, buscando una mejora continua para hacer al sistema más robusto, económico y viable.

1. Aspectos Teóricos

## 1.1 SLAM

Algoritmo para la navegación autónoma en robots móviles. Por sus siglas en ingles Simultaneous Localization and Mapping, de forma literal, el objetivo del algoritmo es hacer que un robot móvil pueda localizarse y/o mapear un entorno al mismo tiempo, para que posteriormente pueda navegar de forma autónoma en el mismo entorno (pues ya se considera como territorio conocido por el robot por ya tener un mapa, localizando cada uno de los obstáculos) sin intervención humana para llegar a su punto final deseado. Este algoritmo se divide en cuatro etapas, Mapeo, Localización, Planificación de caminos y Navegación autónoma.

## 1.2 Odometría

Es el estudio de la estimación y/o suposición (no determinación) de un móvil respecto a la cantidad de movimiento registrado por sus llantas. En el caso de la robótica móvil, los cálculos Odométricos se basan de la lectura de los Encoders que hay en cada una de sus llantas para estimar (por el número de pulsaciones en ambas direcciones, CW o CCW) la distancia recorrida, la dirección del robot y orientación final e inicial del mismo. Particularmente usado para algoritmos de navegación autónoma y/o control a distancia (teleoperación).

## 1.3 Locomoción

La locomoción es el conjunto de mecanismos que usa un objeto para moverse (por su propia cuenta) de un punto A a un punto B. En el caso de los robots móviles, existen tres comunes Sistemas de locomoción, que son, por Ruedas, por Bandas (u Orugas), con Patas y los sistemas Mixtos.

## 1.5 Holonomía

enfocándose en la robótica móvil, mejor se refiera a Sistema Holónomo. Un Sistema Holónomo es aquel que tenga el número de DOF controlables y sea el mismo de DOF que tenga. Lo anterior dicho se traduce que al tener el mismo número de DOF controlables que DOF (si DOFC=3 y DOF=3), el sistema se podrá mover en todas las direcciones posibles tanto de forma longitudinal y lateral. Y si sus DOFC=2 y DOF=3, no puede controlar un DOF, por tanto, su movimiento se restringe en un DOF, o sea, sólo se puede haber movimientos longitudinales, pero no laterales (como un DDMR). Por tanto, un sistema Holónomo puede moverse con libertad y de forma instantánea en cualquier dirección, esto en el sentido más teórico, en la realidad factores dinámicos restringen la palabra instantáneo.

## 1.5 DDMR

Drive Mobile Robot (Robot Móvil de Tracción Diferencial). Robot Móvil con características no Holónomas, con movilidad igual a uno y direccionalidad a dos, con maniobrabilidad de tres, con un comportamiento cuasi Holónomo. Con un mínimo de dos llantas direccionables y una caster Wheel de apoyo como mínimo.

## 1.6 Sensor Activo

Un sensor Pasivo logra detectar una señal (y/o fenómeno) del exterior y responder ante la presencia de este (a determinada intensidad del fenómeno). Y un sensor activo se diferencia principalmente de un pasivo por poder emitir la misma señal con la que trabajará, captarla por reflejo y dependiendo de la información de esa señal captada, dar una interpretación al usuario (como distancia, color, profundidad y forma, etc).

## 1.7 ROS

Robotic Operating System, es un middleware o metasistema operativo con objetivo de ser usado en tareas de robótica, tanto manipuladores como móviles. De Licencia BSD (libre para ser usado en investigación y fines comerciales). Con tareas como simulación de sensores, dinámicas, diseño de estructuras, protocolos de comunicación, visión artificial, aprendizaje autónomo, etc. En el caso del proyecto actual se usará la versión ROS1, programado para su uso en Linux Ubuntu.

1.8 VNC

Por sus siglas en inglés Virtual Network Computing, es un programa para el control de las acciones de una computadora desde otro ordenador a distancia.

# Círculos con flechas contorno2. Etapas de funcionalidad del Sistema.

## 2.1 Locomoción

* La Raspberry pi4 es el elemento principal de cómputo- Estará conectada por i2c a cada una de las tarjetas Arduino, y será monitoreada y controlada por VNC con una Computadora remota conectada a la misma red wi-fi que la Raspberry.
* La parte de locomoción necesita de dos motores con funcionamiento independiente uno del otro, y un encoder incremental de dos direcciones cada uno, y serán controladas por un microcontrolador, en este caso el microcontrolador será una tarjeta Arduino.
* La parte fundamental de esta lograr hacer teleoperación, o sea que, con el “Nodo maestro” corriendo en la Raspberry pi4, activando VNC en la computadora remota y lanzando desde ahí el “Nodo de teleoperación”, y por medio del teclado (de la computadora remota), se deberá controlar los movimientos del robot.

Diagrama

Descripción generada automáticamente

## 2.2 Mapeo

1. Para la etapa de mapeo se añadirá un dispositivo al esquema anterior, y ese será el sensor LiDAR (igual controlado con ROS), el cual también ocupará un puesto USB del microprocesador,
2. Para que sea exitoso el Mapeo, se deberá tener buen control por teleoperación y activado el nodo de Mapeo en la computadora remota.

Diagrama

Descripción generada automáticamente

## 2.3 Planificación de caminos y Navegación Autónoma

Las últimas etapas pueden ser vistas como una sola, como un complemento:

1. Hecho una vez el mapa, se puede correr el nodo de Navegación Autónoma.
2. Se abrirá una interfaz con la interpretación del mapa, se podrá escoger dónde se quiere llegar.
3. Escogido un punto final, se trazará una sería de curvas por dónde la computadora calcula (según las configuraciones que se hizo) la mejor trayectoria que el robot debe realizar.
4. Pero de aquí la razón que se divida esta etapa, pues se necesita analizar las trayectorias hechas por la computadora, y de no ser convincentes, se deben cambiar las configuraciones del sistema de Navegación Autónoma.

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

Si las curvas son satisfactorias, entonces:

1. Se debe monitorear las acciones del robot, tanto velocidad como aceleración, y finalmente, la posición y orientación final.
2. Y si la velocidad y aceleración no son eficientes, deben ser cambiadas las configuraciones del Sistema.

Diagrama

Descripción generada automáticamente

# Ordenador con relleno sólido3. Aspectos del sistema

## 3.1 Hardware

1. 1 Laptop o Computadora remota.
2. 1 Pantalla con entrada HDMI o Display touch para Raspi 4.
3. 1 Cable HDMI.
4. 1 Adaptador HDMI-v8, o mejor llamados, Adaptador micro HDMI
5. 1 Mouse USB.
6. 1 Teclado USB.
7. 1 Fuente de alimentación y sistema de recarga (batería).
8. 1 Raspberry pi 4 Model B.
9. 1 Fuente de alimentación 5.1 V/ 3 A para Raspi 4 (de poder, sea la oficial por parte de Raspberry pi4).
10. 1 Memoria Micro sd HC de 32GB con adaptador.
11. 2 Tarjeta Arduino Nano (como mínimo).
12. 1 Sensor LiDAR YDLiDAR X4.
13. 2 Sensores encoder (mecánicos u ópticos) KY 040.
14. 2 Módulos puente H para control de motores, BTS 7960.
15. 2 Motores RS 550 con caja de motorreducción.
16. 2 Llantas controlables de 20 cm.
17. 2 Llanta de apoyo (Caster o Loca).
18. 2 Ejes de acero inoxidable 5/8” y 25 cm de longitud.
19. 2 Chumaceras de piso de 5/8”.
20. 1 Impresora 3D.
21. 1 Rollo PLA.

## 3.2 Software

1. RaspBian Buster versión 2020-02-14.
2. ROS (en su versión ROSBerry for Raspi 4).
3. Arduino para sistemas Linux armhf.
4. Configuración de los dos buses de i2c de la Raspi 4.
5. VNC (que ya viene instalado en el SO RaspBian Buster).
6. VNC Viewer para la PC remota.

# Interfaz de la experiencia de usuario con relleno sólido4. Preparación del Sistema

## Instalación de Raspbian

1.- Para la descarga del sistema Operativo Raspbian Buster 2020-02-14, se deberá ir al siguiente link y descargar el archivo “2020-02-13-raspbian-buster.zip” del siguiente link: <https://downloads.raspberrypi.org/raspbian/images/raspbian-2020-02-14/>.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

2.- Descargar el Raspberry pi Imager para instalar la imagen del SO en una tarjeta sd. En el siguiente link se encuentra <https://www.raspberrypi.com/software/> .  
  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

La instalación es sencilla, sólo corriendo el ejecutable y dándole “aceptar” y dar los permisos necesarios, estará listo en menos de 10 minutos.

3.- Insertar la sd en la computadora, y esperar a que esta reconozca el dispositivo de almacenamiento. Abrir el programa Raspberry pi Imager.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

1. En “CHOOSE OS”, al darle click se abrirá un navegador del sistema de la computadora, buscar el archivo “2020-02-13-raspbian-buster.zip” en dónde se haya guardado y darle click.
2. Posterior, en “CHOOSE STORAGE” al clickear se abrirá una lista de los dispositivos de almacenamientos disponibles para escribir el SO, en este caso se tendrá que escoger el dispositivo sd.
3. Finalmente, si todo va bien, se habilitará la sección “WRITE”, seleccionando esta opción, toca esperar.

Ahora el SO para la Raspi 4está listo.

4.- Sin conectar a corriente:

1. Colocar la memoria en el slot para sd de la Raspi 4.
2. Conectar la Raspi 4 a la pantalla y/o Display touch.
3. Conectar el mouse y teclado.

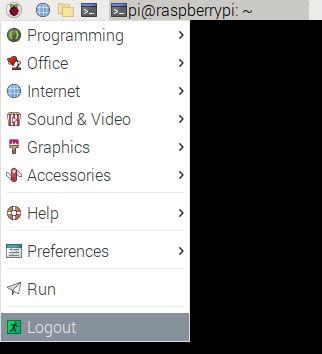
Posterior a estos tres pasos, se está listo para conectar la Raspi a corriente con su fuente de alimentación.

5.- Se verá como carga las configuraciones del SO (por interfaz gráfica) y pedirá que se escoja:

1. Lenguaje del SO.
2. Configuración del teclado.
3. Zona horaria.
4. A las demás opciones, se le debe dar “Continuar” y cuando pida si se quiere actualizar las dependencias del SO, es mejor por ahora no hacerla.

6.- Reiniciar el sistema.

Dando click en el logo de Raspi (una frambuesa) en el lado superior izquierdo de la pantalla.



E ir a la sección “Logout”.

Y de las opciones que despliega, escoger “Reboot” o “Reinciciar”.

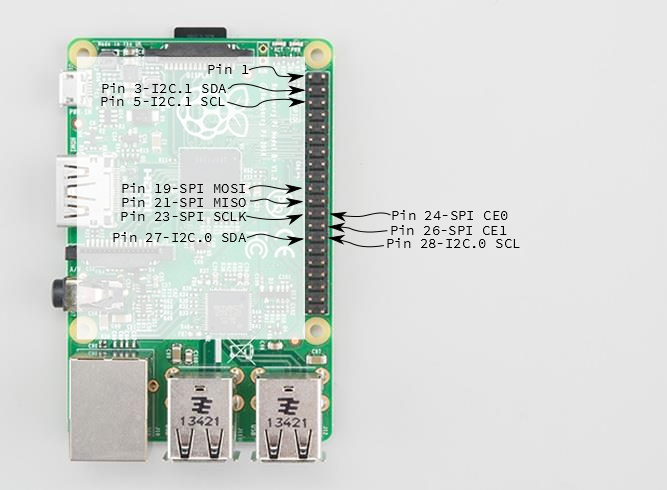


7.- Reiniciado el sistema, se debe actualizar los repositorios y/o dependencias del SO:

1. Presionar ctrl+t par abrir la terminal.
2. Teclear el comando “sudo apt-get update” y las listas de repositorio y/o dependencias están ya actualizadas.
3. Teclear el comando “sudo apt-get upgrade”. Y dar los permisos, que siempre se piden, cuándo se pregunta “¿Si se quiere seguir con la descarga?”, tecleando “s” (en español) o “y” (en inglés).
4. Esperar a que todo se descargue e instale.
5. Una vez terminado, reiniciar el de nuevo.

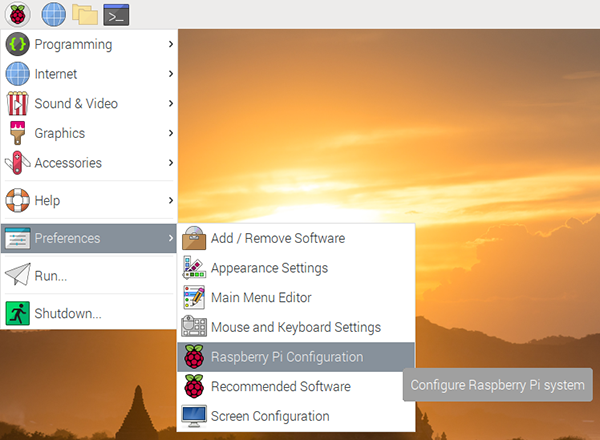
## 4.2 Configuración de los Buses i2c de la tarjeta Raspberry

Primero antes que todo, una tarjeta Raspi 4 tiene 2 buses de i2c. En los GPIO’s 3 y 5 (Bus 1) y en los 27 y 28 (Bus 0).

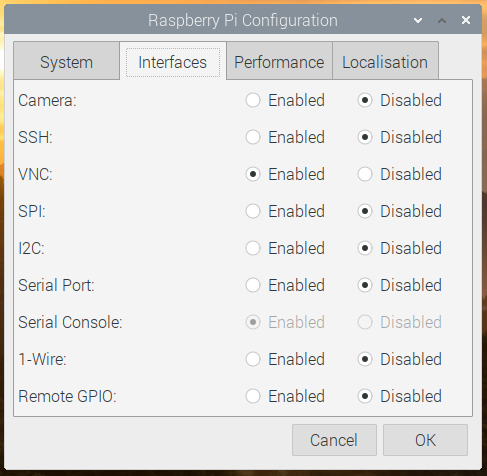


Es especial esta tarjeta, porque en anteriores sólo se contaba con una i2c, en excepción de Raspi 3 que puede tener ambos buses pero no son muy estables. Retomando la Raspi 4, siempre que se habilita la interfaz i2c, sólo se tiene acceso al Bus 1. Y a pesar de que los pines 27 y 28 son predeterminados para conexión EEPROM, igual pueden ser usados para i2c. Empezando con ello:

1. Primero se debe abrir las configuraciones del sistema.



1. Ir a la sección “Interfaces”



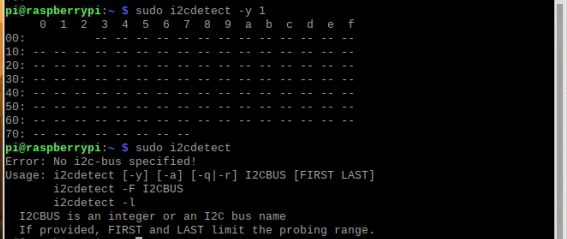
1. Habilitar “I2C” (y para no acceder a esta parte posteriormente, igual habilitar “VNC” y “SSH”)
2. Y darle “OK” en la parte inferior derecha
3. Ahora pedirá reiniciar la tarjeta.
4. Tardará un poco, pero una vez hecho ya estará instalado

* i2c-tools
* VNC Server
* SSH

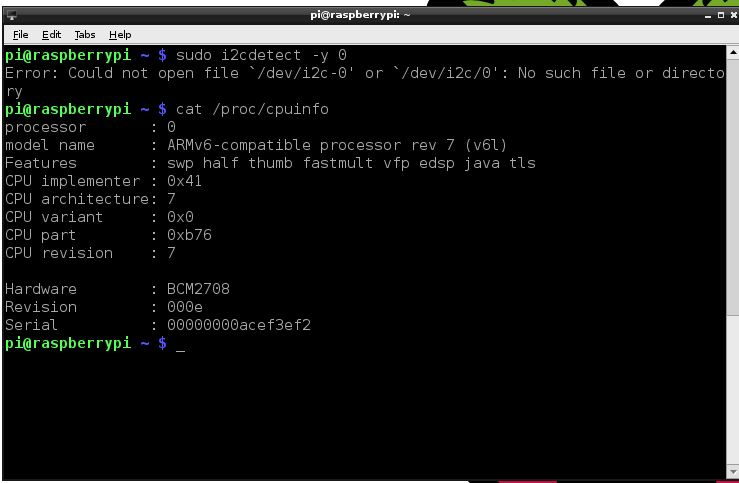
Para verificar que esta instalado las herramientas para la comunicación i2c:

Abrir una terminal nueva.

Teclear “sudo i2c-detect -y 1” (con el comando “i2c-detect” se inspecciona los dispositivos que están conectados, y “1” porque se está viendo por el estado del Bus 1), se debe ver algo así:



Lo cual indica que el Bus 1 este habilitado, pero si se intenta con el bus 0 (“sudo i2c-detect -y 0”):



Esto es normal, ya que no es que este descompuesta la tarjeta, si no, que este Bus sólo se puede habilitar haciendo una modificación en el código fuente del SO, el proceso es el siguiente:

1. Abrir una terminal.
2. Escribir “sudo nano /boot/config.txt”.
3. Se debe ir al fondo del documento, o, rastrear las partes de comunicación i2c.
4. Y justo en esa sección, escribir “dtparam=i2c\_vc=on”.
5. Presionar ctrl+x para salir y guardar los cambios.
6. Si se intenta ahora verificar si el Bus 0 está habilitada, seguirá bloqueado.
7. Reinicie la tarjeta, ahora si, estará el Bus 0.

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

## 4.3 Instalación de ROS

La versión que se instalará es la versión Melodic, si se quiere ver la fuente del articulo para la descarga se deja disponible en <http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Melodic%20on%20the%20Raspberry%20Pi>. Pero en resumen, las acciones a realizar son:

1.- Abrir una Terminal

2.- Instalar las Llaves de los repositorios:

**$** sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

**$** sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

3.- Actualizar los paquetes del SO

**$** sudo apt-get update

**$** sudo apt—get upgrade

4.-Instalar las dependencias Bootstrap mediante

**$** sudo apt install -y python-rosdep python-rosinstall-generator python-wstool python-rosinstall build-essential cmake

5.- Inicializar las Dependencias de ROS, ROSdep

**$** sudo rosdep init

**$** rosdep update

6.-Crear un espacio de trabajo (workspace)

**$** mkdir -p ~/ros\_catkin\_ws

**$** cd ~/ros\_catkin\_ws

7.- Instalar la versión de escritorio de ROS MELODIC

**$** rosinstall\_generator desktop --rosdistro melodic --deps --wet-only --tar > melodic-desktop-wet.rosinstall

**$** wstool init src melodic-desktop-wet.rosinstall

8.- Resolver las dependencias, sin salir de la carpeta ros\_catkin\_ws

**$** rosdep install -y --from-paths src --ignore-src --rosdistro melodic -r --os=debian:buster

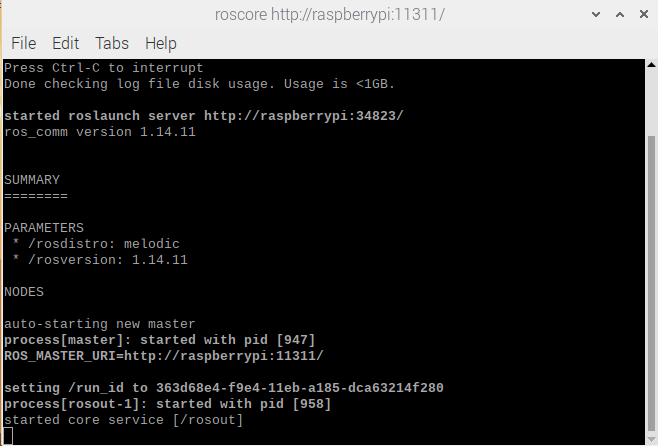
9.- Construcción ROS, la parte más ardua es al construir el programa. Sin salir de la carpeta “ros\_catkin\_ws”:

**$** sudo ./src/catkin/bin/catkin\_make\_isolated --install -DCMAKE\_BUILD\_TYPE=Release --install-space /opt/ros/melodic .

El proceso tardará aproximadamente 1 hora. Se debe poner a ROS en fuente, en bash con:

**$** echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

10.- Verificar que ROS este instalado con escribir en terminal roscore y debe verse como:



11.- Finalmente se debe crear un espacio de trabajo y compilarlo con catkin.

**$** mkdir –p ~/catkin\_ws/src

**$** cd ~/catkin\_ws

**$** catkin\_make.

Al finalizar este proceso se verá que se crearán tres directorios en la carpeta.

Imagen que contiene Texto

Descripción generada automáticamente

- Para ponerlo en el bash permanentemente:

**$** echo “source ~/catkin\_ws/devel/setup.bash”>>~/.bashrc

## 4.4nstalación de dependencias necesarias para los módulos de ROS

Una vez ya instalado ROS Melodic desktop, todas las herramientas, librerías y programas están listas. Pero para tareas de SLAM se necesitan aún instalar más complementos. Las dependencias y/o librerías son:

|  |  |  |
| --- | --- | --- |
| Nombre | Uso | Lista completa de Dependencias |
| ROS communication  ros\_comm | Provee las librerías para la comunicación y herramientas de introspección gráfica. Tales como roscpp, rospy, rostopic. | **$** rosinstall\_generator ros\_comm navigation teleop\_twist\_keyboard ros\_control ros\_controllers geometry slam\_gmapping --rosdistro melodic --deps --wet-only --tar > melodic-custom\_ros.rosinstall |
| navigation | Toma la información de Odometría para crear una aproximación acerca de la posición de la base móvil del robot. |
| teleop\_twist\_keyboard | En robots móviles, controla el PWM y/o velocidad de los motores, con el uso del teclado, simulando un joystick. |
| ros\_control | Rehace los controladores de velocidad, posición y trayectoria del robot PR2 para otros robots genéricos. Tomando como entrada información de posición de articulaciones, velocidad de las llantas y puntos a llegar en el espacio y hacer el control (como PID) y monitorear las salidas. |
| ros\_controllers | Es un complemento para ros\_controll, proveyendo herramientas de control más variadas. |
| geometry | Toma información tal como ángulos, distancias lineales y rotaciones entre articulaciones y la base del robot (y sus respectivas uniones), para el cálculo de transformaciones y la aproximación de posiciones de las mismas articulaciones respecto a la base. |
| slam\_gmapping | Da la posibilidad de realizar el algoritmo SLAM basado en laser. |

Los pasos que seguir son los siguientes:

1.- Ir a la dirección raíz de ROS:

**$** cd ~/ros\_catkin\_ws

**$** rosinstall\_generator ros\_comm navigation teleop\_twist\_keyboard ros\_control ros\_controllers geometry slam\_gmapping --rosdistro melodic --deps --wet-only --tar > melodic-custom\_ros.rosinstall

2.- Ahora, se actualizará el Workspace con wstools:

**$** wstool merge -t src melodic-custom\_ros.rosinstall

**$** wstool update -t src

3.- Para prevenir fallas en la instalación, se inspeccionará si se necesitan dependencias en el sistema, y de ser necesario, se instalarán:

**$** rosdep install --from-paths src --ignore-src --rosdistro melodic -y -r --os=debian:buster

4.- Finalmente, se construirá (instalará) todas las dependencias, el proceso será tardado, ya que de nuevo se construirá cada una de las librerías de ROS. El proceso tardará aproximadamente una hora y media (o incluso dos horas), se recomienda el uso de disipadores de calor para la Raspi y el ventilador especial para este. El comando final es:

**$** sudo ./src/catkin/bin/catkin\_make\_isolated --install -DCMAKE\_BUILD\_TYPE=Release --install-space /opt/ros/melodic

## 4.5 Descarga de los repositorios

Para la descarga de los archivos completos del proyecto:

1.- Clonar los repositorios:

**$** git clone

2.- Mover las dos carpetas del directorio “MD/scr/” al directorio “catkin\_ws/src”

3.- Abrir una terminal

4.- Cambiar la dirección:

**$** cd catkin\_ws

5.- Compilar el proyecto de ROS

**$** catkin\_make

Para más información del tipo de archivos que hay en el repositorio completo y del cómo aportan al proyecto, léase el README del repositorio, de preferencia desde la página de GitHub.

## 4.5 Instalación del entorno Arduino para Raspberry pi4

1.- Ir al siguiente link:

<https://www.arduino.cc/en/software>

2.- Este hipervínculo lo llevará a la página oficial de Arduino, en su sección de software, de todas las opciones de descarga se debe escoger la versión Linux ARM 32bits. Guardarlo en la carpeta más conveniente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

3.- El archivo guardado se deberá descomprimir. Para ello con click derecho y escoger “descomprimir aquí”. Y se tendrá una carpeta llamada “arduino-versión- armlinux” (versión es una forma de generalizar la versión actual en la que se haga estas instrucciones, a la fecha que se escribió esto la versión era la 1.8.19) Dentro de esta abra otro directorio, “arduino-versión”, en este finalmente, estarán todo y cada uno de los archivos para la instalación y desinstalación del IDE Arduino.

4.- Desde el directorio “arduino-versión- armlinux”, dar click derecho y dar en “abrir terminal” y con ello se abrirá una terminal con la dirección actual:

usuario@usuarioPC: ~/Descargas/ arduino-versión- armlinux$

5.- Dar los permisos al directorio con:

**$** sudo chmod -R 777 arduino-version/

Se le pedirá la contraseña de usuario para dar los permisos.

6.- Finalmente, para instalar el IDE se debe ir a la dirección del archivo de instalación y ejecutarlo, escribiendo los comandos:

**$** cd Arduino-version/

**$** sudo ./install.sh

Y ahora la IDE estará lista para ser usada.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

## 4.6 Instalar librerías para Arduino.

Las librerías adicionales necesarias para correr el script de Arduino son PinChangeInt y Arduino-PID-Library, las cuales ayudarán a establecer las configuraciones de las interrupciones y de un control PID, respectivamente. Los pasos son:

1.- Ir a los links:

<https://github.com/GreyGnome/PinChangeInt>

<https://github.com/br3ttb/Arduino-PID-Library>

2.- Ir a la sección “Code”, y del menú que se despliega, escoger “Download ZIP”

Captura de pantalla de computadora

Descripción generada automáticamente

3.- Asignarle una ubicación de guardado, y abrir la dirección donde se guardaron.

4.- Abrir la IDE Arduino, y en la barra Programa >> Incluir Librería >> Añadir Biblioteca .ZIP.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

5.- Escoger el archivo .zip de cada librería y dar en “Abrir”.

# Engranajes con relleno sólido 5. Aspectos de diseño del Sistema de Locomoción y de Odometría

## 5.1 Sistema de Locomoción

Los aspectos y piezas principales para el SL (Sistema de locomoción) son:

1. Con un peso de 10.8 kg.
2. Dos ruedas de caucho de 200 mm de diámetro.
3. Dos ruedas de Castor de caucho.
4. Dos ejes de acero inoxidable de 5/8” y 25 cm de longitud cada uno.



Para fines de explicación se dividirá en secciones el desarrollo del SL.

1.- Se instalaron dos tablas de 5 cm de ancho para fijar una chumacera de 5/8” en cada tabla. Chumaceras que servirán para asegurar la una buena transmisión de movimiento.

Imagen que contiene gato, interior, puerta, pequeño

Descripción generada automáticamente

2.- Para sincronizar el giro de cada motor con las llantas, se diseñó una pieza “Ranura” para impresión en 3D para encajar en la ranura de los motores, la inserción de los ejes de acero y un orificio de 6mm transversal para fijar la Ranura y el Eje.

Una caricatura de una persona

Descripción generada automáticamente con confianza media

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

3.- Para nivelar los motores se hizo la siguiente pieza

Imagen que contiene estructuras metálicas

Descripción generada automáticamente

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

4.- Las piezas impresas en PLA son las siguientes

Un par de zapatos sobre un piso de madera

Descripción generada automáticamente con confianza baja

Imagen que contiene interior, tabla, lavabo, hombre

Descripción generada automáticamente5.- Uniendo las piezas impresas con los motores y ejes:

6.- Instalando todo en el robot, el SL queda listo.

Imagen que contiene interior, tabla, pequeño, cocina

Descripción generada automáticamente

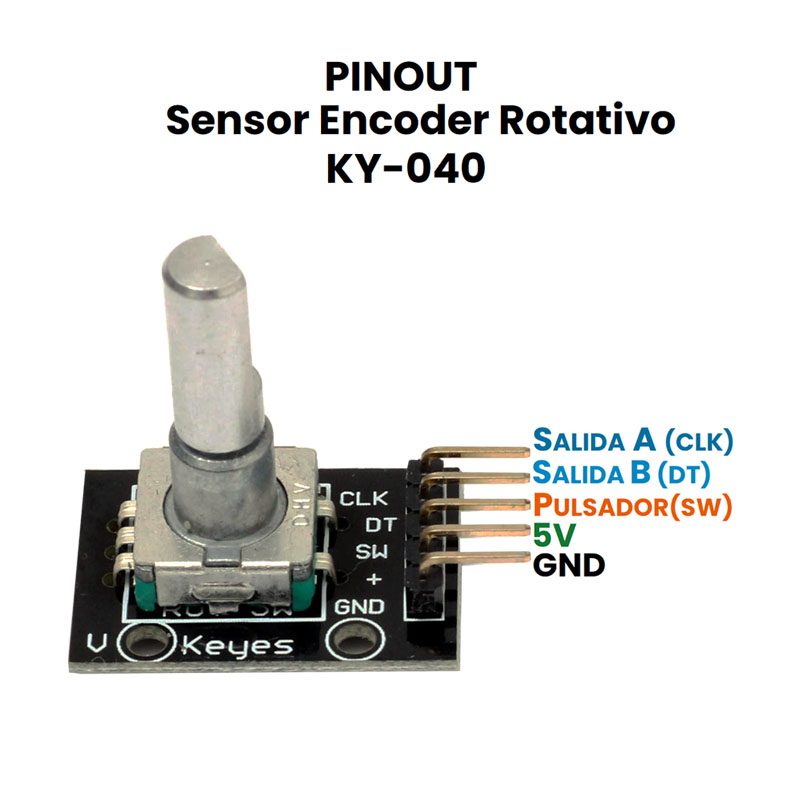
Imagen que contiene puesto, acostado, estacionado, tabla

Descripción generada automáticamente

## 5.2 Sistema de Odometría

EL Sistema de Conteo (SC) consiste principalmente en la implementación del sensor encoder KY-040 en el SL. Los detalles de diseño y el proceso de implementación son:

1.- Con la elección del sensor y el SL finalizado:

Imagen que contiene puesto, acostado, estacionado, tabla

Descripción generada automáticamente

2.- Para fijar el sensor a la estructura del robot se diseñó una carcasa:

Forma

Descripción generada automáticamenteImagen que contiene Forma

Descripción generada automáticamente

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

3.- Las principales medidas para diseñar un Mecanismo de transmisión de movimiento:

31

21

26

**[mm]**

**Chumacera de 5/8”**

**Tabla**

**Encoder con carcasa de PLA**

4.- Se decidió usar engranajes para la transmisión de movimiento del eje de acero hacia la perilla del sensor

Icono

Descripción generada automáticamente con confianza mediaForma

Descripción generada automáticamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Módulo | Dp | N | Grosor |
| Engrane | 4 | 80 mm | 20 | 10 mm |
| Piñón | 4 | 48 mm | 12 | 10 mm |

5.-La simulación del ensamblaje:

**Sensor con carcasa de PLA**

**Eje de acero 5/8”**

Icono

Descripción generada automáticamente

6.- Impresas las piezas e instalados en el robot:

Imagen que contiene interior, tabla, hecho de madera, equipaje

Descripción generada automáticamenteImagen que contiene interior, tabla, escritorio, pequeño

Descripción generada automáticamente

Una maleta de viaje

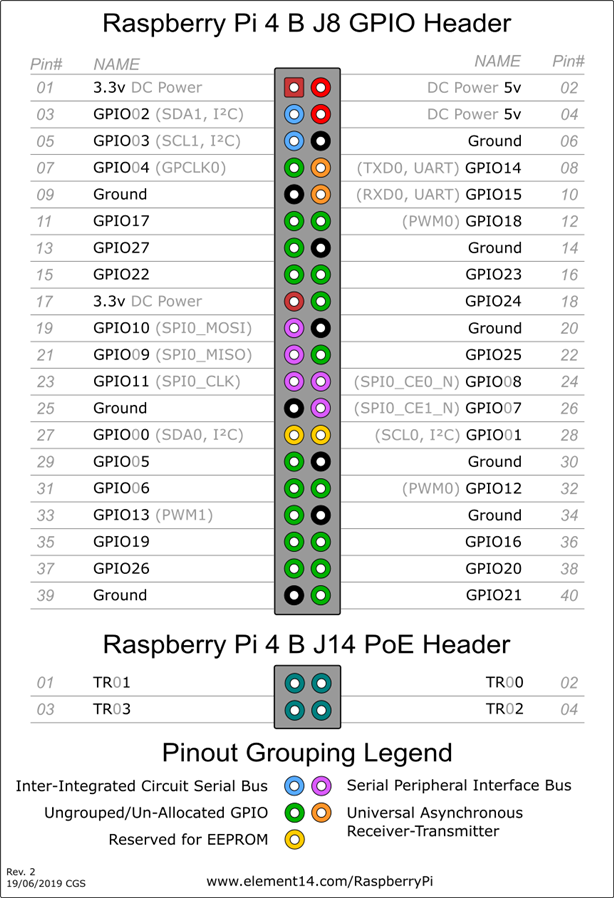
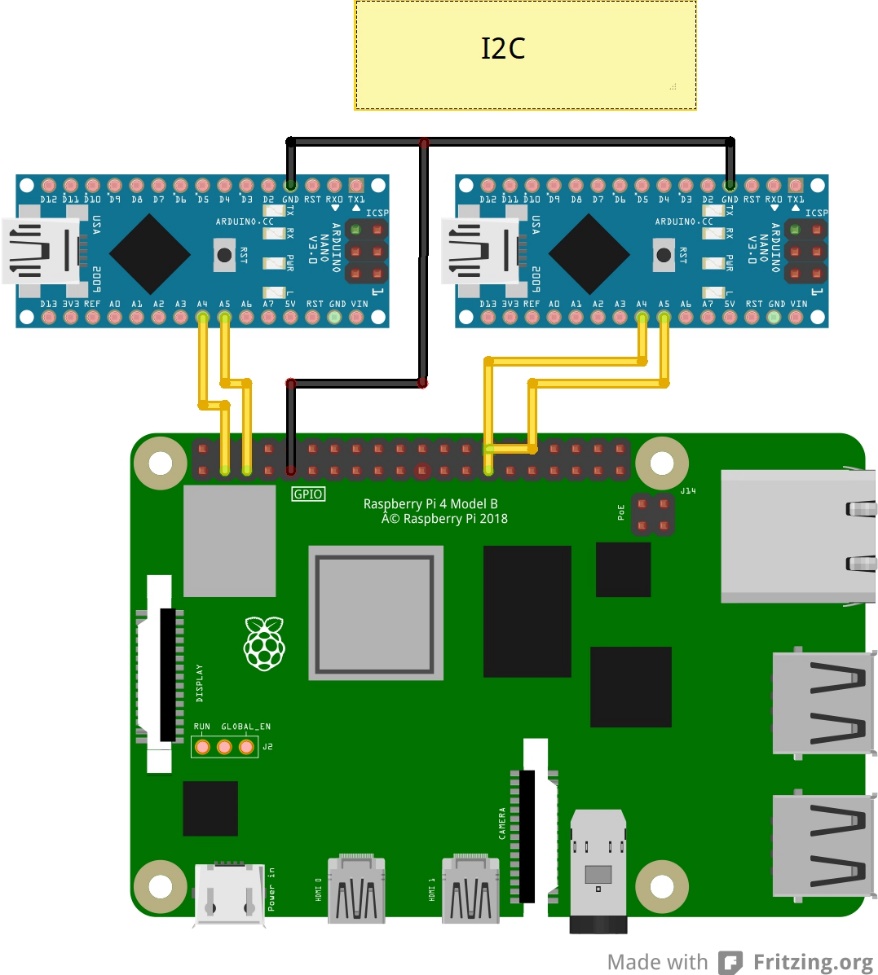
Descripción generada automáticamente con confianza media

# USB con relleno sólido6. Esquema de Conexiones

A continuación, se muestra el código de colores que se siguió para las conexiones y explicación de los diagramas eléctricas y electrónicas.

|  |  |
| --- | --- |
| SEÑAL | COLOR |
| + 12 VCD |  |
| GND, -12VCD |  |
| 5 VCD |  |
| SEÑAL ANALÓGICA |  |
| SEÑAL DIGITAL |  |
| MOTOR 1 |  |
| MOTOR 2 |  |

## 6.1 Conexión para establecer comunicación I2C



## 6.2 Conexión Eléctrica para el Sistema de Locomoción

Diagrama

Descripción generada automáticamente

## 6.3 Conexión Eléctrica para el Sistema de Conteo

Diagrama

Descripción generada automáticamente

## 6.4 Diagrama de conexiones para el Sistema de Navegación

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

# 7. Códigos y programas

# 8. Corrimiento del Sistema