[[1]](#footnote-1)

PROYECTO N° 3

LABERINTO ROBOT RATON

Moncayo Rojas Luis Miguel, Murcia Juan David, Pérez Guzmán Daniel

[Lmmr.530@gmail.com](mailto:Lmmr.530@gmail.com)

[juancia96@gmail.com](mailto:juancia96@gmail.com)

[ingdani7@gmail.com](mailto:ingdani7@gmail.com)

Universidad Del Quindío, Facultad de ingeniería, Armenia Quindío.

*Resumen*— para el desarrollo del proyecto, se pretende realizar el diseño de un laberinto, el cual el robot ratón tendrá como objetivo resolver lo en menor tiempo posible, para ello se implementara la simulación en tiempo real en la herramienta computacional processing que permitirá visualizar cada uno de los movimientos que realizar el robot ratón

*Índice de Términos—* Micro controlador, sensor qtr 8, puente H L298N, processing.

# INTRODUCCIÓN

El presente informe trata de la construcción de un robot ratón capaz de resolver un laberinto. Para ello primero se empieza con la construcción del laberinto de dimensiones 6\*6, cada cuadro tiene una medida de 25 cm, una vez diseñado el laberinto, el robot raton será capaz de detectar las paredes para evitar chocar, también será capaz de detectar cual es el camino más adecuado y rápido para llegar a su punto de destino.

En particular, para la elaboración del presente proyecto se utilizó una placa Arduino Uno y sensores para medir los valores de la pista, junto con la herramienta computacional processing con la cual se pretende crear una interfaz el cual permitirá en tiempo real observar la ubicación del robot ratón.

# objetivos

* realizar un robot ratón con la capacidad de resolver un laberinto con el menor tiempo posible, evitando quedar encerrado o colisionar con alguna de las paredes.
* Diseñar y construir una estructura para el laberinto
* el robot tendrá la capacidad de ilustrar en una interfaz gráfica cada uno de los cuadrantes que avanza en su recorrido

# MATERIALES

* Puente H
* 2 motores DC
* QTR-8
* Madera(estructura del robot seguidor de línea)
* Dos ruedas locas
* Arduino uno
* Dos baterías de 8 voltios
* Cables
* Protoboard
* Laberinto

IV MARCO TEORICO

1. PUENTE H L298N

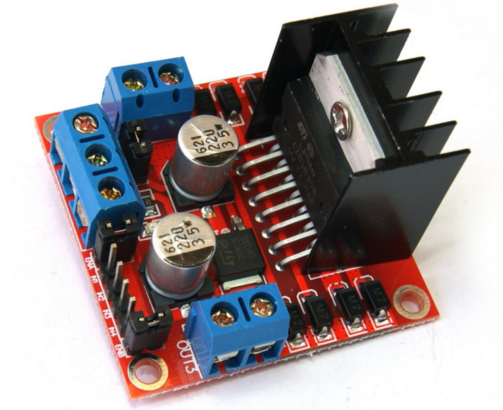


Figura 1. Puente H L298N.

Un puente H es un circuito electrónico que permite a un motor eléctrico en DC girar en ambos sentidos, avanzar o retroceder, son ampliamente usados en robótica y como convertidores de potencia [1]

El Puente H L298N se usa para controlar los motores DC y alimentar el micro controlador

1. MOTORES DC 10:1



Figura 2 motor DC

Es una máquina que convierte la energía eléctrica en una mecánica provocando un movimiento rotatorio, gracias a la acción que se genera del campo magnético [1]

Los motores se usa para dar el movimiento al robot seguidor de línea

1. SENSORES QTR-8

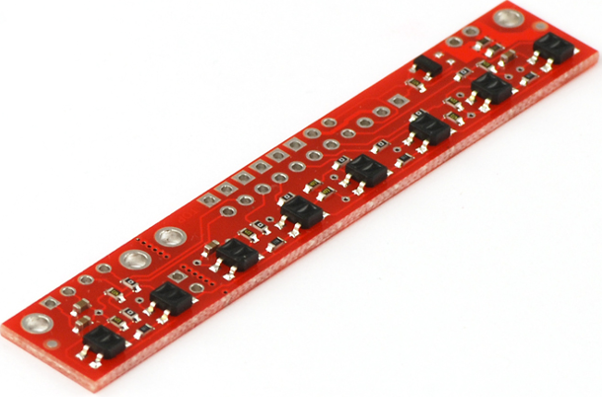


Figura 3 sensores QTR-8

Cada sensor proporciona por separado una salida de voltaje análoga. Se pueden utilizar 8 sensores

El arreglo sensor de reflexión QTR-8 está pensado como un sensor de línea, pero puede ser utilizado para identificar la proximidad o como un sensor de reflectancia. El módulo dispone de ocho emisores de infrarrojos y de 8 receptores (fototransistor). El fototransistor se conecta a una resistencia de pull-up para formar un divisor de voltaje que produce una salida de voltaje analógica entre 0V y VIN, en función del IR reflejado. Bajo voltaje de salida es una indicación de una mayor reflexión. [2]

1. ARDUINO UNO

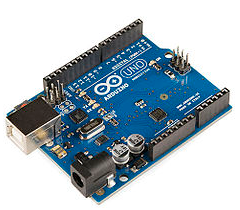


Figura 4 arduino uno

 Es una [placa](https://es.wikipedia.org/wiki/Circuito_impreso) de desarrollo que integran un [micro controlador](https://es.wikipedia.org/wiki/Microcontrolador) y un [entorno de desarrollo](https://es.wikipedia.org/wiki/Entorno_de_desarrollo) (IDE), diseñado para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El [hardware](https://es.wikipedia.org/wiki/Hardware) consiste en una placa de circuito impreso con un micro controlador, usualmente [Atmel AVR](https://es.wikipedia.org/wiki/AVR), y puertos digitales y analógicos de [entrada/salida](https://es.wikipedia.org/wiki/Entrada/salida), los cuales pueden conectarse a placas de expansión que expanden las características de funcionamiento de la placa arduino.

1. RUEDA LOCA



Figura 5 rueda loca

La rueda loca es la que va a guiar el carro en las curvas y rectas, se encuentran ubicadas en la parte de adelante del carro

1. BATERIAS



Figura 6 baterías

La batería es la fuente de energía para propulsar el robot seguidor de línea

1. LLANTAS



FIGURA 7 llantas seguidor de línea

Las llantas del robot son las que van a mover el robot seguidor de línea y son movidas por los motores, se usan unas llantas grandes para que el robot pueda andar más rápido.

1. Microcontrolador

Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada y salida

# DISEÑO, PROCEDIMIENTO Y ANÁLISIS

Diseño

* Primero se realizó el plano de elaboración del laberinto el cual ilustra la imagen 8

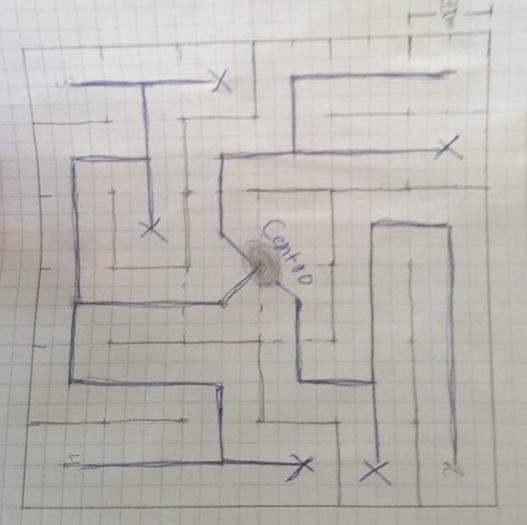


Figura 8 bosquejo laberinto

* Primero se elaboró un laberinto con dimensiones de 1,50m\*150m, dividiendo cada uno de sus cuadrantes en 25\*25 cm de esta forma se obtiene una matriz cuadrada de 6\*6.

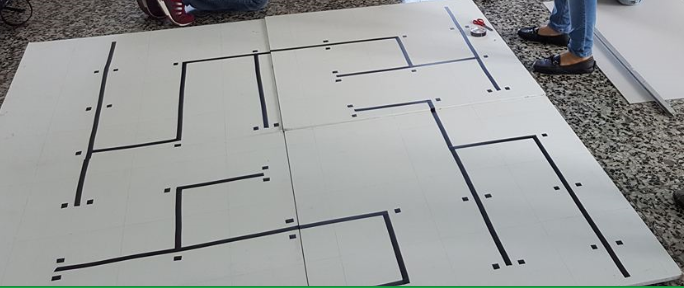


Figura 9 diseño y elaboración laberinto

* luego se trabajó en el algoritmo para resolver el laberinto, el código implementado se ilustra en la figura 10

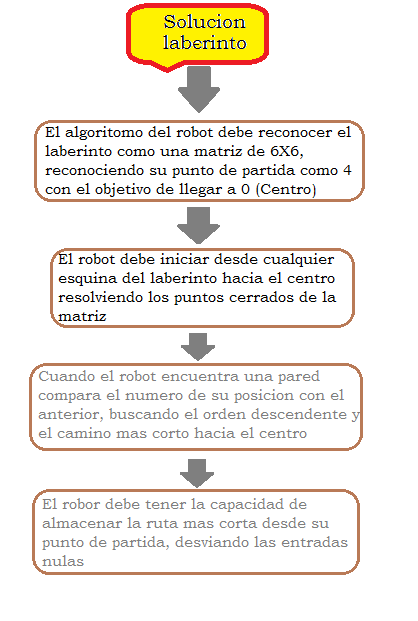


Figura 10 diagrama de bloques del algoritmo implementado en el robot ratón

La realización del laberinto ha sido diseñado en proccesing con el fin de simular el entorno de trabajo de robot y visualizar la solución de laberinto en la interfaz. La figura 11 ilustra el laberinto en proccesing

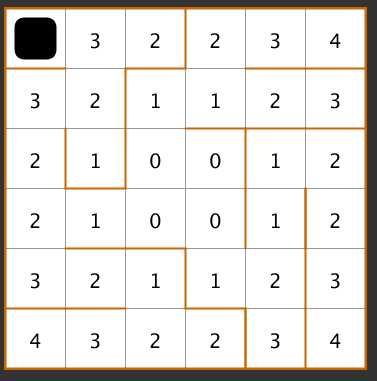


Figura 11 laberinto en proccesing

La programación se realiza en el lenguaje c, java y c++ como es una programación orientada a objetos se trabaja con la clase MAXE que ha sido diseñada por los ingenieros Gerardo Galoge Lopez y Luis Miguel Capacho la cual contiene herramientas que facilitan la implementación del algoritmo, esta cuenta con las siguientes funciones.

* getvalue:

Getvalue permite tomar el valor de la matriz en la que está ubicada el robot

* setvalue:

Setvalue permite actualizar el valor de ccada uno de los cuadros que avanza el robot, esta es muy aplicada en el autofill

* setwalls:

Setawalls permite la ubicación de paredes entre celda ya sea en cualquier ubicación del borde de la celda enviándole la celda a la que se le desea implementar paredes y eb que parte se desean ubicar utilizando las definiciones NORT, SOUTH, EAST, WEST, para las direcciones .

* .getwalls:

Getwalls permite conocer los valores de donde se encuentran las paredes y así solucionar el laberinto.

* .display:

Esta función permite visualizar en la interfaz la posición del robot en el laberinto.

* En la figura 12 se puede observar el diagrama de flujo del algoritmo de funcionamiento del robot ratón.

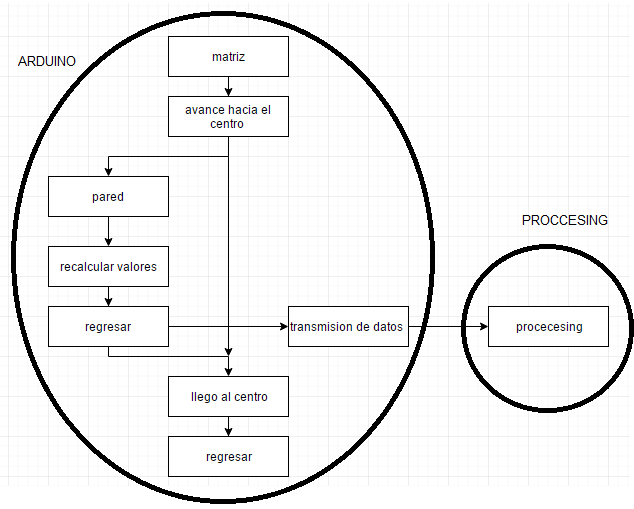


Figura12. Algoritmo diagrama de bloques del ratón.

# CONCLUSIONES

* Basándose en el esquema del robot seguidor de línea del proyecto pasado se optó por recortar su estructura para facilitar su regresión en caso de encontrar un objeto
* La herramienta computacional processing permite visualizar cada uno de los cambios de cuadrantes que realiza el robot, estos son leídos por el sensor QTR.
* Para el caso cuando el robot está reconociendo inicialmente el laberinto es necesaria que la velocidad de los motores sea baja ya que si es muy alta no alcanzará a reconocer el obstáculo más cercano, cuando el robot tiene almacenado el recorrido más corto hacia su destino los motores están al cien por siento de velocidad para llegar lo más rápido posible a su destino
* Para establecer la comunicación entre el robot y la interfaz de processing es necesario un módulo de bluetooh en el Arduino, este permitirá la comunicación micro – pc
* En caso de no contar con el sensor QTR y el laberinto no contara con las líneas negras y con los indicadores de cambio de cuadrante sería necesario
* que el robot contara con sensores de proximidad

# BIBLIOGRAFÍA

[1] <https://es.wikipedia.org/wiki/Puente_H_(electr%C3%B3nica)>

[2] <http://tdrobotica.co/sensor-de-linea-qtr-8a-analogo-arreglo-de-8/173.html>

[3]<http://aprendiendofacilelectronica.blogspot.com.co/2014/12/robot-velocista-de-competencia_4.html>

1. [↑](#footnote-ref-1)