Robot solucionador de laberintos mediante algoritmo flood fill

Andres Felipe Ardila. Cód.1094954068

Diego Leonardo CárdenasCód.1094955953

Maycol Sabogal Ardila Cód.1115192075

Microprocesadores. Programa de Ingeniería Electrónica, Facultad de Ingeniería, Universidad del Quindío - Colombia

***Resumen –Este proyecto se llevo a cabo bajo la implementación previa de un robot seguidor de línea el cual fue modificado para darle paso a un robot capaz de resolver un laberinto usando el algoritmo flood fill aunque dando prioridad a su funcionamiento inicial, es decir, el robot modificado tiene la característica de seguir la línea pero se le añade una función especial para que se encargue de resolver el laberinto, como también modificar su capacidad de giro en 90 y 180 grados***

***Palabras clave –Microcontrolador, Arduino, laberinto, reconocimiento,***

# **INTRODUCCIÓN**

En el mundo de la ingeniería es necesario saber adaptarse a cambios inesperados y poderse enfrentar a ellos teniendo la máxima capacidad de reacción hacia esto. En este proyecto se partió de un robot seguidor de línea el cual tenía como fundamento el control PID y fue necesario modificarlo para que este fuera capaz de resolver un laberinto, aprovechando al máximo las funciones del dispositivo ya construido.

A partir de las ya mencionadas características del robot inicial y agregando nuevas funciones como su capacidad de movimiento lineal y en giros de 90 y 180 grados, también la modificación en el uso de los sensores QTR encargados de seguir la línea, al detectar un cambio de cuadrante en el laberinto lo cual permite darle memoria y en un segundo intento hacerlo de una manera más eficiente. En base a esto se construyo un algoritmo capaz brindarle las nuevas funciones al dispositivo partiendo de su desempeño primario.

# **OBJETIVOS**

## **General:**

Implementar correctamente un robot seguidor de línea capaz de resolver laberintos desconocidos, usando las plataformas de processing y arduino conjuntamente.

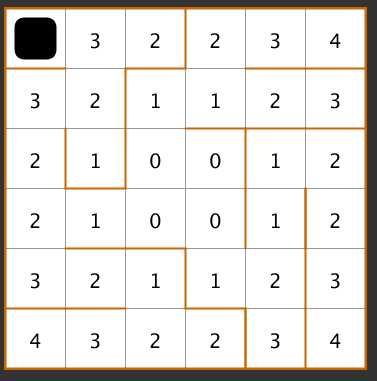
***Específicos:***

* Desarrollar un código funcional para la solución de laberintos con algoritmo flood fill en la plataforma arduino.
* Realizar de manera óptima una interfaz en el software processing para la visualización de movimiento y actualizaciones de datos en el laberinto.
* Optimizar recursos, para así tener un robot de bajo costo y fácil configuración.

# **MARCO TEÓRICO**

**Algoritmo Flood Fill**

El algoritmo flood fill determina el área por elementos contiguos de una matriz de orden n\*n, la cual se divide en sub cuadriculas que son el posterior registro de un posible robot. El algoritmo rastrea las cuadriculas por las cuales no deberá transitar nuevamente añadiendo a ella una etiqueta y guardando en un registro su camino. Este algoritmo puede valerse de la representación matricial de las celdas del laberinto lo que es factible a su vez representarse eficientemente en la memoria de un sistema digital como una computadora o un microcontrolador, de esta manera el algoritmo de control representará características como la posición y la orientación mediante vectores de números.

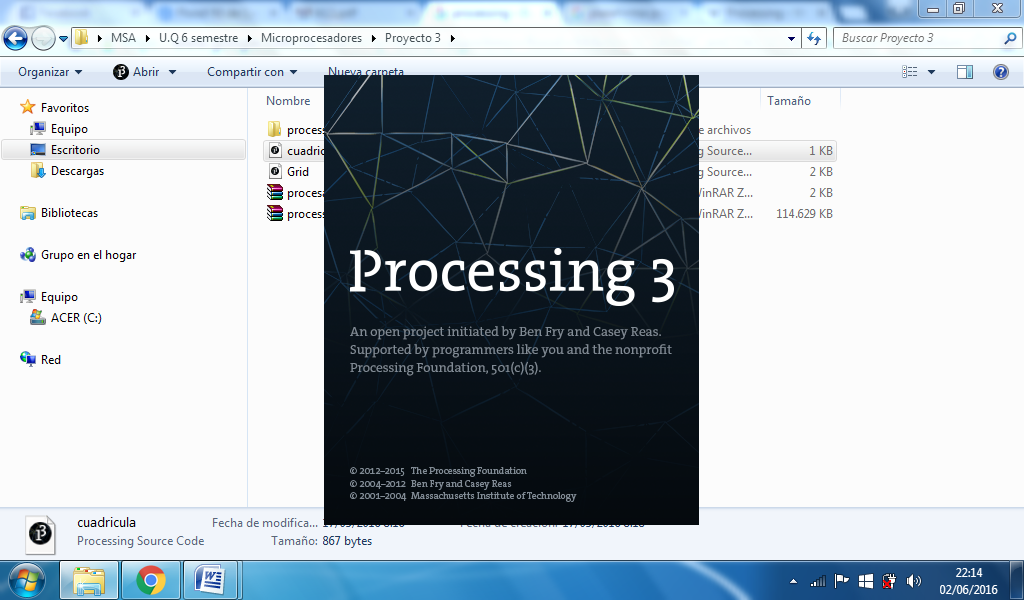


*Figura 1. Cuadricula base de algoritmo flood fill.*

**Processing**

Es una plataforma de programación y entorno de desarrollo integrado basado en Java, esta plataforma es una herramienta de fácil utilización y tiene gran aplicación en el desarrollo de proyectos multimedia e interactivos de diseño digital.

Al estar basado en Java, puede heredar todas sus funciones convirtiéndose en una herramienta poderosa a la hora de encarar proyectos complejos.



*Figura 2.Sotfware Processing.*

# **DISEÑO Y PROCEDIMIENTO**

Para la realización del proyecto 3 de microprocesadores se realizó el siguiente procedimiento:

* Primero se tomó como base el proyecto dos el seguidor de línea para el cual se le anexo que pudiera resolver un laberinto aplicándole un flood fill
* Para ello al robot se le carga la cuadricula en forma de matriz mostrada en la figura 1 sin incluir la posición de las paredes ya que el carro las tendrá que identificar al realizar su primera vuelta
* Para la implementación del flood fill en arduino primero se definiéndole una posición de meta que para nuestro caso se escogió la 3 en x y la 2 en y el carro avanzara hasta que se encuentre con una pared que será cuando todos los sensores de la Qtr obtengan un valor menor a 20 ese fue nuestro valor máximo obtenido al calibrar la regleta en blanco
* Al recibir este valor los motores del carro se detendrán y en la posición actual le sumara 1 luego el carro girara 90 grados hasta hallar una línea y poder seguirla y de ahí se le sumara 1 también a todas las posiciones que ya ha recorrido con eso se logra rellenar de nuevo la matriz con los valores corregidos para así en su segundo recorrido el carro tomara el camino más corto ya encontrado al centro
* La posición del carro se le estará enviando por medio de comunicación inalámbrica al puerto del pc y de ahí a la plataforma de Processing la cual permite observar el comportamiento del carro en una interfaz grafica

# **CONCLUSIONES**

* Mediante el montaje básico de un robot seguidor de línea se puede adaptar un solucionador de laberintos, sin agregar componentes, realizando el código debido.
* La plataforma de processing es muy útil para este tipo de proyectos, puesto que al trabajar con lenguaje de programación java es muy orientada a objetos y por ende se pueden visualizar interfaces graficas que varían en tiempo real con el cambio de parámetros en un montaje físico.
* Mediante el campo de la electrónica se pueden realizar interesantes proyectos de robótica, los cuales se pueden aplicar a muchas cosas cotidians para solucionar determinados problemas.
* El algoritmo de flood fill es muy eficiente para su aplicación en desarrollo de laberintos para robots que cumplen este propósito.

# **web grafía**

[1]<https://es.wikipedia.org/wiki/Algoritmo_de_relleno_por_difusión>.

[2] <https://www.pololu.com/docs/0J19/all>

[3] https://processing.org/reference/