AVANCE PROYECTO ROBOT RESUELVE LABERINTOS

Para la implementación del proyecto se tomara parte del anterior trabajo como lo es la parte física del montaje y la utilización de los módulos reconocedores de la línea negra, se incluirá una parte que detalla la solución que el carro le proporciona al laberinto en donde se visualizara el mapeo en un programa externo al de arduino, este tiene como objetivo interpretar lo que llega del puerto serial de arduino y mostrarlo dinámicamente en una interfaz personalizada, el programa que muestra esta ilustración se denomina Processing y basa su funcionamiento en Java un lenguaje de programación al igual que C. El proceso que se ha llevado acabo, en primera medida es el rediseño total y parcial del mini-auto que es el encargado a través de su chasis de llevar los componentes de funcionamiento y la alimentación, el vehículo como tal se hizo más compacto y pequeño tomando medidas muy precisas, esto se realizó con la finalidad de que no interfiriera con el camino del laberinto denotando su correcto giro sin chocar con las paredes y pudiendo así lograr una reducción de posibles errores a la hora de realizar el trayecto.

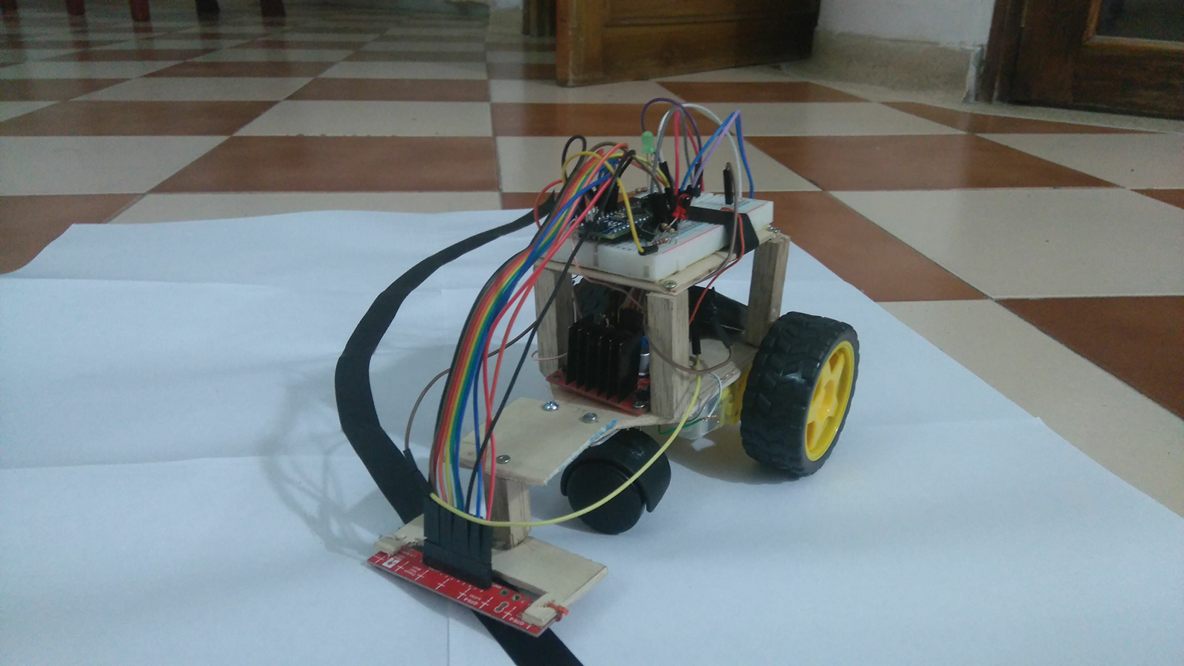


**Fig. 1** motores anteriores



**Fig. 2** motor nuevo a usar

Debido a las fallas con los motores de la fig. 1 se optó por conseguir motores de mejor calidad que albergan más movimiento igualitario y proporcional (ver figura 2), también se reemplazó los sensores hechos a mano desde cero por un módulo completo que incluía 8 sensores y una librería que facilitaba su uso (Pololu Qtr), el carro como tal quedo de nuevo hecho y se evidencia en la fig. 3 su montaje.



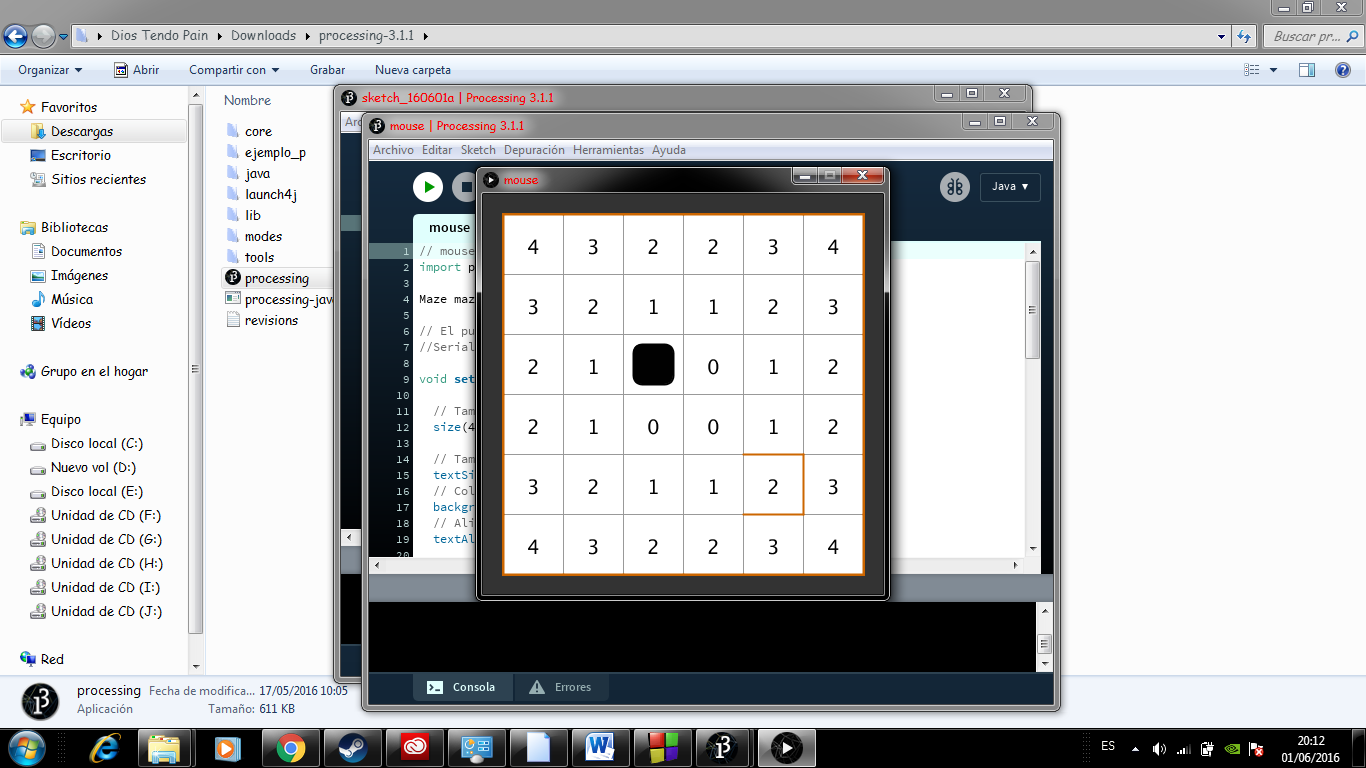
**Fig. 3**  carro finalizado, listo para pruebas

Como tal el código fue reemplazado en su totalidad debido a que la lógica nuestra difiere en gran medida en la que proporciona el sensor, ya que este basa su funcionamiento en la librería propia del módulo (la librería se anexara en el repositorio), las pruebas de giro aún están siendo realizadas y cuando se tenga algo contundente será anexado.

La lógica que seguirá el carro es simple, primero se sabrá que la matriz será de 6x6, el vehículo iniciara en la posición (0,0) siendo esta la primer parte coordenada de la matriz, el carro deberá buscar el camino descendiente más cercano para esto se le asigna valores que descienden de 1 en 1, cuando el valor es ascendente el carro no debería seguir ese camino (esto ocurre porque hay un muro que le impide seguir), una vez haya dado con el centro del laberinto tendrá que recorrer el laberinto sin choques y sin meterse en caminos inciertos. Al final y durante el proceso debe haber comunicación serial entre el computador y el arduino para constatar el correcto mapeo en el Processing.

El vehículo se supone que no debe saberse el laberinto y por lo tanto a medida que lo soluciona debe graficar en la interfaz de la figura 4.

De la figura 4 también se visualiza una pared que fue puesta a modo de prueba para verificar la gráfica resultante de las paredes



**Fig. 4** cuadricula a llenar

Ahora, se analizaran las funciones establecidas en el código proporcionado por el profesor Luis Miguel Capacho

**Setwalls:** Esta función correlaciona las diferentes ubicaciones entre celdas y utilizando la predefinición NORT, EAST, SOUTH, WEST se direcciona el lado que se quiere dibujar.

**Setvalue:** esta función relaciona el valor donde está ubicado el seguidor de línea, modificándolo.

**Getvalue:** getvalue es una función que extrae el valor donde se encuentra el seguidor de línea en el laberinto

**Getwalls:** esta función se caracteriza por obtener los valores donde se encuentran las paredes y así gracias a ello poder dar solucionar al laberinto. Esta función retorna un valor numérico potencia de 2 (8 si es WEST, 1 NORTH, 4 SOUTH, 2 EAST)

**Display:** es la función encargada de dibujar la cuadricula a utilizar dándole parámetros como el tamaño y el color, además es la encargada de dibujar el mouse que simula la posición del carro en la matriz.