

Vehículo de entregas inteligente

(20 Septiembre 2020)

Garcia Solares, Gonzalo Antonio 201318652
 Ordoñez Xiloj, Ronald Geovany 201314564
 Paredes Sol, Walter Josue 201504326
 Sinay Alvarez, Mynor Styven 201403520

Resumen— El presente proyecto consiste en la realización de un vehículo inteligente, debido a la situación actual que tenemos en el país por la pandemia de COVID-19 se ha ideado la forma de poder llevar cada uno de nuestros paquetes sin que nosotros nos tengamos que movilizar a nuestro buzón de espera con la finalidad de no tener contacto alguno, para esto también se contara que el carrito sea eficaz, es por eso que seguirá un camino el cual contendrá una serie de obstáculos que deberá detectar durante todo su recorrido, para poder seguir su camino de forma eficaz contara con sensores seguidores de línea en lo que podrá regresar a su estado de origen una vez recogido el paquete que deberá de ser pesado durante el recorrido.

I. NOMENCLATURA

Arduino, sensores, bitácora, ultrasónico, API-REST, base de datos, seguidor de línea.

II. INTRODUCCIÓN

Este documento proporciona la elaboración de un buzón inteligente, el cual estará hecho de cartón piedra reforzado el buzón contara con una puerta manual para que se pueda abrir para poder depositar nuestro pedido, al momento de recibir nuestro pedido, se nos mostrara una notificación en nuestro dispositivo móvil en el cual nos indicara que tenemos un objeto dentro de nuestro buzón, así mismo el peso de dicho objeto, se procederá a la desinfección de nuestro pedido por medio del rociador de líquido, nuestro buzón nos notificara de cuanto es líquido que disponemos por una serie de colores para que procedamos a rellenarlo de liquido y no quedarnos sin desinfectante

III. HARDWARE

- Vehículo
- Microcontrolador Arduino
- Modulo Wi-fi
- Protoboard
- Motores DC

IV. SENSORES

- Sensor de ultrasónico: Utilizaremos este sensor con la finalidad de que podamos determinar la cantidad de obstáculos que habrá durante el recorrido del carrito .



Fig 1: Sensor ultrasónico

- Sensor MD-SP2: Utilizaremos este sensor para poder detectar que tanto peso tenemos dentro de nuestro carrito cada vez se introduzca un paquete nuevo



Fig 2: sensor MD-SP2

- Sensor CNY70: se utilizará este sensor para poder seguir la línea que será trazada en el piso el cual indicara la ruta del carro en función.

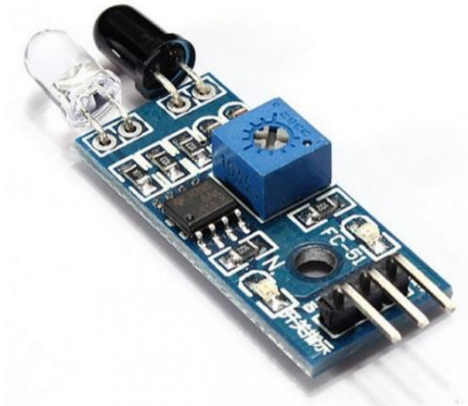


Fig 3: sensor CNY70

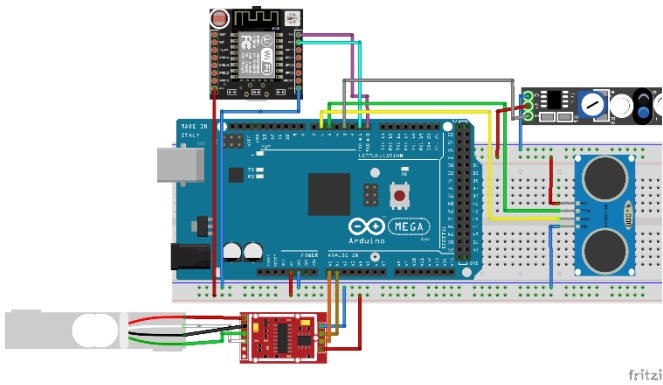


Fig 4: Conexión de Arduino con los sensores

V. PROTOCOLO DE COMUNICACIÓN

El vehículo utiliza comunicación Wifi para mandar la recolección de los datos a una API, por lo que también está implícito el uso de tecnología API-REST, al utilizar formato JSON para el intercambio de datos y el uso de métodos estándar POST, GET, etc. Luego la API se comunica con la base datos de Mongo para almacenar los datos que recibe, y la aplicación móvil se comunica vía internet con la API la cual le retorna los datos que le solicita para su visualización.

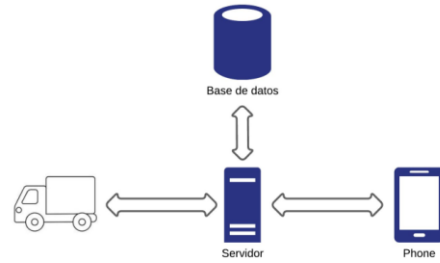


Fig 5: Arquitectura a utilizar

VI. ANÁLISIS

Cada uno de los datos serán almacenados en una base de datos el cual está conectado con la API-REST para poder hacer cada una de las peticiones necesarias, por un lado tenemos el seguir de línea que deberá de ser capaz de poder seguir su camino de manera directa para poder llegar con el usuario final a su destino, en el camino habrá una cantidad de obstáculos, los cuales el vehículo podrá detectarlos para poder llevar el contador de cuales fueron los que encontró durante toda su trayectoria, ya que llevara un peso considerado este debe de tener la fuerza suficiente para poder transportarlo de manera correcta, lo cual una vez entregado el paquete deberá de regresar a su posición inicial.

VII. APP

La aplicación consistirá en una pantalla la cual se podrá observar un dashboard en donde podremos monitorear todos los aspectos de nuestro vehículo, tales como: ubicación, estado, cantidad de paquetes entregados, los obstáculos que se va encontrando en el trayecto del recorrido, un aproximado del tiempo que se tardó en cada entrega, la aplicación contará con notificaciones las cuales serán de tipo push, esto significa que se mostrarán con el resto de las aplicaciones, teniendo en consideración que podremos poner de manera manual los diferentes modos como lo son activos e inactivo.

Salida		
Peso	Hora y fecha	Obstáculos
0	0	0

Llegada		
Peso	Hora y fecha	Obstáculos
0	0	0

Regreso		
Peso	Hora y fecha	Obstáculos
33	2020-09-27T01:07:58.703Z	20

Dashboard		
Ubicación	Estado	Paquetes entregados
Regreso a base	Inactivo	1

Total		
Peso	Tiempo X	obstáculos
33	0	20

Activo
Inactivo
MODOS

Fig 6, interfaz de la aplicación

VIII. CONCLUSIONES

Se logro cumplir con cada uno de los requisitos solicitados, pero asi mismo uno de los principales problemas que se tuvieron fue la calibración del sensor seguidor de línea y dado que no están aislados los receptores este siempre seguirá su camino aun no habiendo una línea que seguir, esto se logró solucionar con la implementación de una “casita” para poder aislar toda la luz ambiente para que pueda ser funcional, otro de los principales fue con la fuerza de los motores dado que la batería de descargaba a la mitad de la trayectoria, pero aun así con esos percances se lograron completar cada uno de los apartados solicitados.

IX. ANEXOS

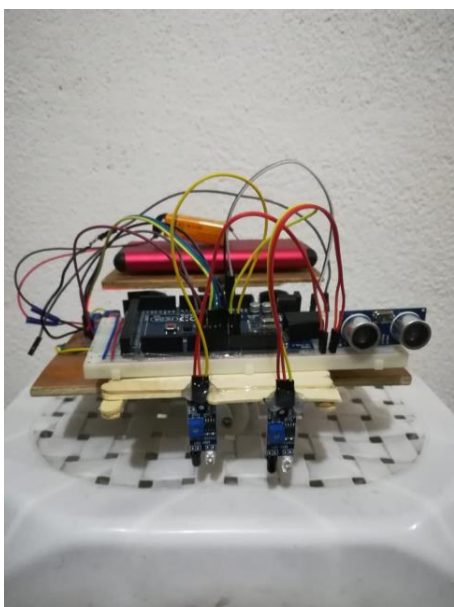


Fig 7: prototipo del vehículo

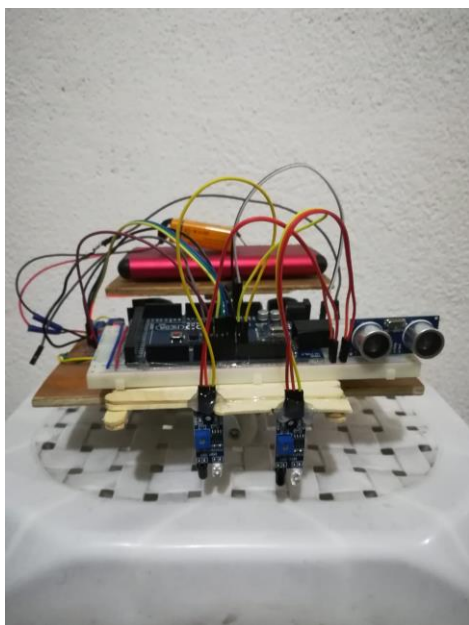


Fig 8: prototipo del vehiculo con los sensores implementados.

REFERENCIAS

- [1] https://www.wika.es/landingpage_weighing_technology_es_es.WIKA
- [2] <https://www.mongodb.com/es>
- [3] <https://www.taloselectronics.com/blogs/tutoriales/celda-de-carga-hx711>
- [4] <https://www.youtube.com/watch?v=LaMaLE0rtGQ>
- [5] <https://loopback.io/doc/en/lb4/>
- [6] <https://hetpro-store.com/TUTORIALES/cny70-sensor-optico/>