

**ADRIAN MORAN DE LA TORRE**

**SANTIAGO REYNAGA ARAMBULA**

**KEVIN RICARDO MARCHENA CORONADO**

**ING. DE SOFTWARE**

**VEHICULO ANTI-CHOQUE**

**INTRODUCCION**

El proyecto de un carrito antichoque controlado por Arduino es una emocionante aplicación de la tecnología que busca mejorar la seguridad y reducir los accidentes en el ámbito de los vehículos de pequeña escala, como los carritos de juguete o los proyectos de robótica educativa. Mediante la combinación de componentes electrónicos, sensores y programación, este proyecto permite crear un sistema inteligente capaz de evitar colisiones y proteger el carrito y sus alrededores. En este proyecto, el Arduino se convierte en el cerebro del carrito, siendo responsable de recopilar datos de los sensores, procesarlos y tomar decisiones rápidas en función de la información recibida. Los sensores utilizados en el sistema pueden incluir sensores de ultrasonido, infrarrojos o de proximidad, los cuales tienen como objetivo medir la distancia entre el carrito y los obstáculos circundantes. Cuando el sensor detecta un obstáculo cercano, envía esta información al Arduino, que utiliza algoritmos y lógica programada para determinar la acción adecuada a tomar. Dependiendo del diseño y los componentes utilizados, el Arduino puede activar los motores del carrito para cambiar de dirección, detenerlo por completo o realizar una maniobra evasiva para evitar la colisión. Este proyecto no solo brinda una experiencia práctica en la programación y la electrónica, sino que también permite comprender los principios básicos de los sistemas de seguridad y la toma de decisiones automáticas. Además, ofrece la posibilidad de explorar y experimentar con diferentes configuraciones y ajustes para optimizar el rendimiento del carrito antichoque.

**ALCANCE**

El alcance de este proyecto de un carrito antichoque controlado por Arduino se define en las siguientes etapas y limitaciones:

**Diseño y montaje del sistema**: El proyecto incluirá el diseño y montaje físico del carrito antichoque, que estará equipado con los componentes necesarios, como el Arduino, sensores, actuadores y otros elementos electrónicos requeridos para su funcionamiento.

**Selección y configuración de los sensores:** Se seleccionarán los sensores adecuados, como sensores ultrasónicos, infrarrojos o de proximidad, en función de los requisitos del proyecto. Se llevará a cabo la configuración de los sensores, calibrándolos correctamente y estableciendo rangos de detección adecuados.

**Programación del Arduino:** Se programará el Arduino para procesar los datos de los sensores, implementar algoritmos de detección y toma de decisiones, y controlar los actuadores en respuesta a los obstáculos detectados. Se utilizará un lenguaje de programación compatible con el Arduino.

**OBJETIVO**

El objetivo de este proyecto de un carrito antichoque controlado por Arduino es desarrollar un sistema inteligente capaz de evitar colisiones y proteger el carrito y sus alrededores. A través de la combinación de componentes electrónicos, sensores y programación, se busca lograr los siguientes objetivos:

**Diseñar un sistema de detección de obstáculos:** El proyecto tiene como objetivo implementar sensores adecuados, como sensores de ultrasonido, infrarrojos o de proximidad, para detectar obstáculos cercanos al carrito. Se busca asegurar una detección confiable y precisa para tomar decisiones en tiempo real.

**Desarrollar un algoritmo de toma de decisiones**: El proyecto tiene como objetivo programar un algoritmo en el Arduino que procese los datos recopilados por los sensores y tome decisiones rápidas y efectivas para evitar colisiones. Se buscará establecer criterios claros para determinar las acciones que el carrito debe realizar en respuesta a diferentes situaciones de obstáculos.

**REQUISITOS**

**REQUISITOS FUNCIONALES:**

**REQUISITOS DE INFORMACION DEL SISTEMA**

El sistema deberá almacenar la información en arduino ya que este ejecutara el código y este debe ser capaz de ejecutar la tarea correctamente se utilizarán cables para pasar los datos y los sensores registrar un objeto para poder detenerse correctamente al atravesarse un objeto

**REQUISITOS DE REGLAS DE NEGOCIO DEL SISTEMA**

El código será desarrollado en base a la plataforma de arduino con este es el único lenguaje que se puede utilizar junto con una serie de sensores para poder que el carrito pueda hacer lo que se pide contener los sensores indicados y los materiales ya preestablecidos por el usuario.

**REQUISITOS DE CONDUCTA DEL SISTEMA**

En este caso lo que se desea destacar es que el carrito al detectar un obstáculo este se detenga totalmente con los sensores puestos en su chasis o al cruxarle un obeto a este que este se de tenga la conducta en general del sistema es que el carrito se active mediante el código que se le programara al arduino para que este haga su funcionalidad correcta mente ande y se detenga totalmente y con esto cabe destacar que seria la conducta que el usuario de sea

**REQUISITOS NO FUNCIONALES DEL SISTEMA**

En este caso que el carrito tenga ciertas modificaciones estéticas ya sea alerones cambiar el tipo de material de la carcasa las ruedas incluso modificar el frente para que se vea deportivo el carrito, en cuestión de seguridad que este pueda frenar o desactivar los servo -motores para no arrollar al obstáculo.

**REQUISITOS DE PORTABILIDAD**

Que el vehículo no supere los 30 cm de largo, 15 cm de ancho y 20 cm de alto ya que se busca hacer lo mas practico posible para este prototipo y con ello que tenga una buena apariencia que sea practico y no tan grande ya que se busca ahorrar lo mayor posible.

**REQUISITOS DE SEGURIDAD**

En este caso que el programa tenga configurados los sensores para que al momento de poner un obstáculo que el vehículo apague los motores y con esto cumpla su función de no arrollar los obstáculos.

**RESTRICCIONES TECNICAS DEL SISTEMA**

Desarrollarse en arduino con el lenguaje correspondiente que este sea compatible en cualquier plataforma y en cualquier computadora que tenga los programas correspondientes para que este se pueda cargar correctamente en el dispositivo