

IMPLEMENTACIÓN DE TÉCNICAS ROBUSTAS DE GUIADO DE VEHÍCULOS AUTÓNOMOS BASADAS EN VISIÓN ARTIFICIAL

AUTOR: JAVIER BLANCO ABIA

TUTORES:

ÁLVARO MAGDALENO GONZÁLEZ
ANTOLÍN LORENZANA IBÁN
C. A, I.T, M.M.C Y TEORÍA DE ESTRUCTURAS



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

**IMPLEMENTACIÓN DE TÉCNICAS ROBUSTAS
DE GUIADO DE VEHÍCULOS AUTÓNOMOS
BASADAS EN VISIÓN ARTIFICIAL**

Autor:

Blanco Abia, Javier

Tutor(es):

Magdaleno González, Álvaro

Lorenzana Ibán, Antolín

**Departamento C.A, I.T, M.M.C
y teoría de estructuras**

Valladolid, junio 2024

Resumen

El objetivo de este TFG en ingeniería electrónica industrial y automática es el diseño y desarrollo de un AGV (*Automated Guided Vehicle*) guiado por visión artificial para el transporte de un *shaker* utilizado en pruebas de carga estructural.

El proyecto se fundamenta en el uso de motores brushless y de placas de hoverboard reprogramadas, lo que proporciona una solución económica. Para la navegación autónoma del AGV se emplean técnicas de visión artificial.

El vehículo está diseñado para transportar el *shaker* de un extremo a otro de un puente, facilitando la realización de pruebas de carga en condiciones repetibles, sin depender de la capacidad de los peatones de seguir con precisión un ritmo prefijado.

El proyecto se estructura en varias fases, incluyendo el diseño y desarrollo del sistema de visión artificial, la implementación de un control adecuado de los motores, y la realización de pruebas de funcionamiento y rendimiento.

Palabras clave

AGV, Visión, *Shaker*, Hoverboard, Brushless

Abstract

The objective of this Final Year Project in Industrial Electronics and Automation Engineering is the design and development of an AGV (*Automated Guided Vehicle*) guided by artificial vision for transporting a *shaker* used in structural load tests.

The project is based on the use of brushless motors and reprogrammed hoverboard boards, which provide an economical solution. Artificial vision techniques are employed for the autonomous navigation of the AGV.

The vehicle is designed to transport the *shaker* from one end of a bridge to the other, facilitating load testing in repeatable conditions, without relying on pedestrians' ability to precisely follow a predetermined pace.

The project is structured in several phases, including the design and development of the artificial vision system, the implementation of appropriate motor control, and the performance and functionality testing.

Key words

AGV, Vision, *Shaker*, Hoverboard, Brushless

Índice

CAPITULO 1 INTRODUCCIÓN.....	1
1.1 Descripción general del proyecto	1
1.2 Objetivos	3
CAPITULO 2 ESTADO DEL ARTE	5
2.1 Trabajo previo.....	5
2.2 Introducción a los AGV	7
2.3 Proyectos similares	9
CAPITULO 3 SELECCIÓN DE COMPONENTES	11
3.1 Selección de la cámara	11
3.2 Selección del microcontrolador.....	16
3.3 Selección de las placas y los motores	17
3.4 Electrónica auxiliar.....	19
CAPITULO 4 PRIMEROS PASOS Y CONSTRUCCIÓN	21
4.1 Prototipado	21
4.2 Chasis del AGV	27
4.3 Sistemas de seguridad.....	29
4.4 Integración de la electrónica.....	30
CAPITULO 5 CONTROL Y PROGRAMACIÓN	35
5.1 Reprogramación de las placas y control de los motores	35
5.2 Programación del modo automático.....	45
5.3 Mando de control.....	48
CAPITULO 6 PROBLEMAS ENCONTRADOS Y SOLUCIONES.....	51
6.1 Interferencias electromagnéticas.....	51
6.2 Alimentación de la electrónica de control	52
6.3 Sobrecalentamiento.....	53
6.4 Control limitado de la dirección.....	54
CAPITULO 7 IMPLEMENTACIÓN Y RESULTADOS.....	57
CAPITULO 8 CONCLUSIONES	63
8.1 Conclusiones	63
8.2 Líneas de mejora	65
8.3 Consideraciones adicionales	68
BIBLIOGRAFÍA	73

Referencias y fuentes:.....	73
ANEXOS	75
Anexo A: Diseños 3D.	75
Anexo B: Códigos desarrollados.	75
Anexo C: Esquemas eléctricos.	75
Anexo D: Lista de componentes y estimación de precios.	75

CAPITULO 1 INTRODUCCIÓN

1.1 Descripción general del proyecto

El proyecto se basa en la construcción y control de un AGV de bajo coste guiado por visión artificial, capaz de transportar objetos de gran peso. Para ello se utilizará de base el trabajo de Fernando Fonseca Porras [1].

Se desarrollará, por tanto, un vehículo autónomo de cuatro ruedas cuya misión será el transporte del *shaker* de la figura 1 para la realización de ensayos a lo largo de puentes.

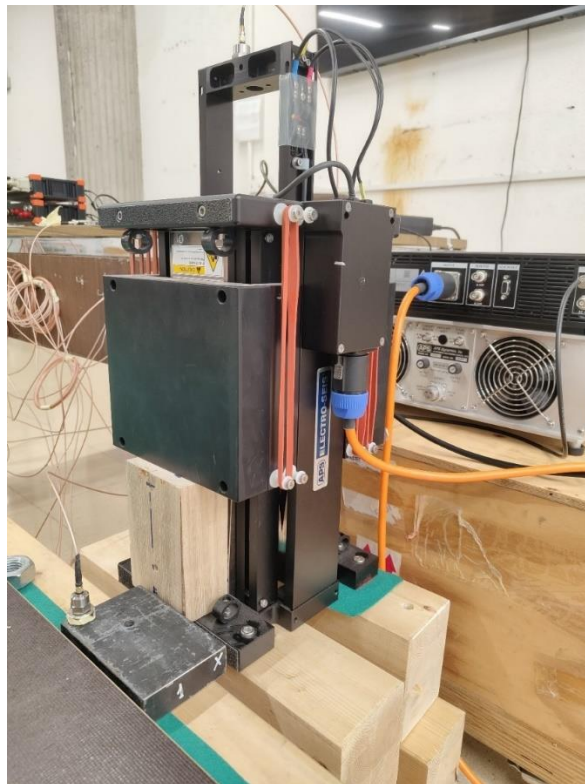


Figura 1. Shaker

El *shaker*, es un instrumento utilizado para pruebas de carga estructural. Dispone de una masa oscilante de la que se sirve para provocar vibraciones a diferentes frecuencias. Esto permite recabar datos sobre la estructura. Existen diferentes modelos de *shaker*, en el laboratorio se dispone de uno de 50 Kg y de otro de 70 Kg aproximadamente.

La necesidad de construcción de un vehículo para su traslado no sólo viene de su elevado peso, el cual ya lo hace difícilmente manejable para una persona. Además, es necesario para la realización de pruebas de carga en condiciones repetibles, sin depender de la capacidad de los peatones de seguir con precisión un ritmo prefijado.

El guiado de la máquina se realizará mediante una de dos opciones, la primera es un control manual remoto, y la segunda a través de una cámara de visión artificial cuyo objetivo es la detección de un cable de un color vistoso y fuertemente diferenciado del entorno, el cual marcará la ruta a seguir.

El cuerpo del robot está conformado por perfiles metálicos y tablas con las sujeciones necesarias para fijar el *shaker*. Podemos verlo en la figura 2.



Figura 2. AGV

La electrónica está basada en placas de hoverboard reprogramadas para un control personalizado de los motores brushless mediante un microcontrolador ESP32 (figura 3), del cual se aprovecharán sus funciones inalámbricas integradas (wifi, radio, bluetooth), para el guiado del robot.

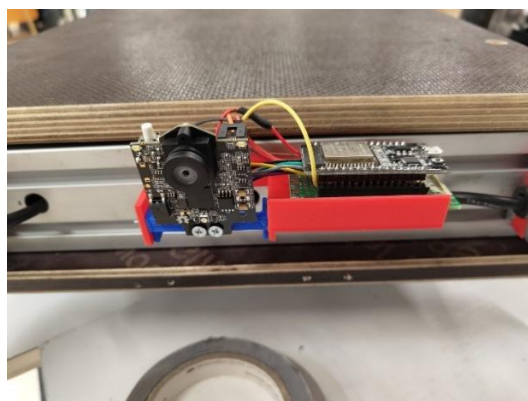


Figura 3. Electrónica de control

1.2 Objetivos

Los objetivos específicos necesarios para la realización de este proyecto son los siguientes:

- Desarrollar un sistema de guiado robusto mediante visión artificial. La ruta establecida debe ser detectada y completada en ambiente de campo, con condiciones de luminosidad y terreno variables.
- Lograr un control satisfactorio de los motores brushless de la máquina. Cuyo requisito indispensable es mantener una velocidad constante y muy reducida, a la vez que un movimiento fluido y con un par suficiente.
El robot debe ser capaz de mover sin problemas el *shaker* el cual es un elemento pesado de unos 50 Kg.
- Realizar una construcción segura y fiable de la máquina, asegurando tanto sus elementos mecánicos como electrónicos. Esto conlleva la integración de los elementos de seguridad, software y hardware necesarios para evitar cortocircuitos, sobrecalentamientos, pérdida de control del robot, enganches etc.
- Diseñar un modo manual remoto que permita la maniobrabilidad y control de la máquina para su preparación y retirada.
- Implementar una configuración sencilla y versátil que permita que cualquier persona sin experiencia previa pueda manejar el robot en su modo manual o realizar los cambios necesarios en la configuración de su modo automático en función del entorno.

Estos objetivos guiarán el desarrollo integral del proyecto, asegurando que cada aspecto crítico del diseño y funcionalidad del AGV sea abordado de manera efectiva. Con ello, se pretende no solo cumplir con los requisitos técnicos, sino también proporcionar una solución práctica y accesible para aplicaciones industriales y logísticas.

CAPITULO 2 ESTADO DEL ARTE

2.1 Trabajo previo

Como se ha mencionado anteriormente, este proyecto es la continuación y mejora del trabajo de fin de grado de Fernando Fonseca Porras, compañero de la escuela de ingenierías industriales. El objetivo y alcance de dicho trabajo son similares a los de este proyecto, la construcción de un vehículo autónomo sigue líneas para el transporte de cargas pesadas.



Figura 4. Chasis AGV original. [1]

Como puede verse en la figura 4, el aspecto exterior del vehículo es prácticamente idéntico al actual. Sin embargo, su planteamiento contaba con algunos fallos o posibles líneas de mejora que se han explorado a lo largo de este proyecto:

- Carencia de elementos de seguridad como seta de emergencia, interruptores para cortar la corriente, fusibles para limitar la tensión etc.
- Sensores inadecuados. Con el fin de seguir la ruta trazada en este diseño se utilizaron sensores infrarrojos de presencia, que, si bien en determinadas condiciones pueden servir para diferenciar el color negro dentro de un entorno controlado, no son adecuados para ese propósito ni para usarse en condiciones de campo con luminosidad variable.
- Pérdida de la documentación sobre la programación del robot. Esto imposibilita la modificación del programa ya que se ha perdido tanto los archivos del *firmware* con el que se han reprogramado las placas como la documentación sobre el programa del ESP32 que controla el movimiento y la comunicación con el mando.
- Deterioro o pérdida de piezas clave en el robot. Tanto para la sujeción de la electrónica como para sujeción de los motores se diseñaron unas piezas en 3D que no estaban correctamente impresas para ese fin o que se han perdido.
- Control insatisfactorio de los motores, ya que el robot nunca llegó a moverse con fluidez ni a cargar con el *shaker*.

Estos problemas se traducen en que, aunque la construcción del robot llegó a efectuarse, no pudo ser utilizado para su propósito inicial. Este proyecto pretende rehacer el vehículo convirtiéndolo en un prototipo funcional para su posterior uso en un entorno real.

No obstante, hay ciertos aspectos del trabajo previo que no se han modificado. El diseño y construcción del chasis del vehículo se ha mantenido inalterado a excepción de algunos ajustes. Además, a pesar de haber realizado cambios importantes en la electrónica y la programación, buena parte de los componentes mecánicos y electrónicos se han mantenido del proyecto anterior. En el capítulo 3 se profundizará sobre este tema.

2.2 Introducción a los AGV

Los Vehículos Guiados Automáticamente (AGV, por sus siglas en inglés) son robots móviles diseñados para transportar materiales en fábricas, almacenes y otras instalaciones industriales. Operan sin intervención humana directa y siguen rutas predefinidas marcadas por cables, líneas magnéticas, láseres o sistemas de navegación más avanzados como el GPS. Podemos ver a uno en funcionamiento en la figura 5.



Figura 5. AGV industrial. [2]

Historia y evolución

Los AGV han sido una parte esencial de la automatización industrial desde mediados del siglo XX. Su desarrollo y uso se pueden dividir en varias fases:

- 1950s-1960s: Los primeros AGV fueron introducidos en la década de 1950. Eran bastante simples, usando cables enterrados en el suelo para guiarlos a lo largo de rutas fijas.
- 1970s-1980s: En las décadas siguientes, la tecnología de los AGV se expandió con la introducción de sistemas de control más sofisticados, incluyendo sensores y controles programables.

- 1990s-2000s: Los avances en la tecnología de sensores y la informática permitieron una mayor flexibilidad en la navegación y control de los AGV. Se empezaron a usar sistemas guiados por láser y tecnologías de radiofrecuencia.
- 2010s-presente: La incorporación de inteligencia artificial y la integración con sistemas de gestión de almacenes han llevado a los AGV a un nuevo nivel de autonomía y eficiencia. Los AGV modernos pueden operar en entornos dinámicos y adaptarse a cambios en tiempo real.

El precio de un AGV varía significativamente según su complejidad, capacidad y características específicas.:

- AGV básicos: Entre 20,000 y 50,000 USD. Estos suelen ser modelos simples con navegación básica.
- AGV avanzados: Entre 50,000 y 150,000 USD. Incluyen sistemas de navegación más sofisticados y capacidades adicionales como la manipulación de carga.
- AGV personalizados y de alta capacidad: Más de 150,000 USD. Estos son adaptados a necesidades específicas y pueden incluir características avanzadas como navegación autónoma y capacidades de integración complejas.

El coste de implementar un AGV no solo incluye el precio inicial del equipo, sino también los gastos asociados a su instalación, mantenimiento y posible personalización según las necesidades del usuario. Por ello no sólo es necesaria la construcción de una alternativa económica, hay que asegurar que sus prestaciones son adecuadas y tener en cuenta el gasto de instalación a la hora de realizar su diseño.

Los rangos de peso para los que están pensados varían con el modelo, pero en general van desde los 500 Kg hasta los 10000 kg. Este trabajo no pretende obtener un modelo de escala industrial, sin embargo, no existen hoy en día AGV más básicos y baratos que permitan mover con facilidad cargas medias.

En este caso se pretende mover una carga de unos 100 Kg para un laboratorio de investigación de la universidad de Valladolid para el cual, a falta de modelos de estas características en el mercado se debe diseñar y construir uno a medida.

2.3 Proyectos similares

Este proyecto pretende utilizar piezas de hoverboard para construir un AGV de diseño personalizado para las necesidades que se requieren.

Los hoverboard son dispositivos de transporte personal, eléctricos, que generalmente constan de dos ruedas y una plataforma sobre la cual el usuario se coloca (figura 6).



Figura 6. Hoverboard. [3]

Estos vehículos están diseñados para el transporte de personas y por lo tanto están diseñados para soportar cargas de entre 50 y 100 Kg.

Generalmente están conformados por dos motores brushless de gran potencia, una batería de 36 V con 1h de autonomía (en función de la conducción), una placa controladora de los motores y dos giroscopios.

Esto lo hace una fuente de material idónea para un proyecto de estas características. Aunque dado el elevado peso que hay que transportar serán necesarios al menos 4 motores.

La reutilización de componentes de *hoverboard* no solo supone una reducción significativa de costes, sino que también fomenta la sostenibilidad al dar una segunda vida a estos dispositivos. Además, el amplio soporte comunitario y la disponibilidad de recursos en línea facilitan el desarrollo y la solución de problemas que puedan surgir durante el proyecto.

Actualmente existe una comunidad en internet que se dedica a crear proyectos basados en los componentes de los hoverboard y a divulgar información sobre ellos. Existen proyectos como coches autónomos, karts eléctricos, coches teledirigidos, vehículos de movilidad personal etc. En la figura 7 podemos ver un ejemplo de uno de estos proyectos.



Figura 7. Kart eléctrico. [4]

Esto brinda una importante fuente de información a la hora de manipular los componentes de los *hoverboard* para utilizarlos en el proyecto.

Gracias a esta comunidad pueden conocerse aspectos clave como los diferentes *firmware*, el método para reprogramar las placas, la configuración de los pines, aspectos técnicos sobre los tipos de control y sus propiedades etc. Todo ello se encuentra disponible de consulta en la bibliografía [4].

Es importante tener en cuenta la complejidad que supone la reconversión de este tipo de materiales. Actualmente existen numerosos modelos diferentes de *hoverboard*. Todos ellos son muy parecidos pero la naturaleza de sus componentes cambia lo que supone que no todos los *firmware* funcionarán en todas las placas ni el método de reprogramación será similar para todas ellas. En ocasiones algunos de estos componentes pueden ser de mala calidad y ser sumamente delicados. Incluso a veces ni si quiera se respeta el código de colores para cables de potencia, pudiendo ser el positivo y el negativo de cualquier color. Esto hace que los materiales sean difíciles de manipular.

CAPITULO 3 SELECCIÓN DE COMPONENTES

Para cumplir los objetivos propuestos para este trabajo hay una serie de componentes clave que se deben investigar y seleccionar cuidadosamente:

- Es necesaria una cámara y una unidad de procesamiento que trate las imágenes tomadas.
- Unos motores con potencia suficiente para trasladar el *shaker*, así como unos controladores apropiados para ellos.
- Un microcontrolador programable que gobierne los motores.
- Elementos de seguridad que garanticen la integridad de la máquina y de las personas cercanas a ella
- Una fuente de alimentación apropiada, así como la electrónica necesaria para adecuarla a todos los elementos.

3.1 Selección de la cámara

El primer paso es la selección del nuevo sistema de guiado del robot. Para ello es necesario un sistema de visión artificial capaz de detectar con seguridad y precisión un determinado color dentro de la imagen el cual se corresponderá con la línea que marca la ruta.

Existen muchos tipos de sensores que permiten realizar esta función (láser, magnéticos, infrarrojos, capacitivos, ópticos...) cada uno con sus ventajas y desventajas.

Algunos de los sensores más utilizados son los magnéticos que detectan una línea dibujada con una cinta especial, este sistema es caro y lento de implementar al igual que los sensores capacitivos. Los sensores láser y LIDAR son muy eficientes e incluso permiten crear planos de la zona de trabajo, sin embargo, son complejos, poco económicos y tienen más prestaciones de las necesarias. Los sensores infrarrojos son sencillos y económicos pero su utilización en diferentes suelos condiciones de luminosidad pueden afectar a su funcionamiento.

Para la aplicación planteada el AGV debe seguir trayectorias prácticamente rectilíneas de forma muy fiable, en condiciones muy variadas de luz y superficie y de forma económica.

Dado que el dibujo de las rutas variará con cada puente y se realizará en múltiples localizaciones, una opción económica sería desplegar un cable de un color llamativo y diferenciable del entorno. De esta manera un sensor óptico podría detectarlo fácilmente y su despliegue y recogida serían rápidos y sencillos.

Por estos motivos se ha decidido que el sensor que mejor se adecua a las necesidades del proyecto es un sistema de visión artificial.

Para ello se han planteado tres alternativas diferentes:

- Un sistema de visión con cámara y unidad de cómputo que permita procesar los datos de la imagen. Para este planteamiento podría servir prácticamente cualquier cámara, la cual se conectaría a una unidad Raspberry Pi o similar que incorporaría un software personalizado que realizase las tareas necesarias para la detección de la línea (figuras 8 y 9). Este sistema requeriría del desarrollo de código necesario para su detección y su coste sería no sería demasiado reducido ya que los precios de las Raspberry Pi rondan los 50 € a lo que habría que sumar el precio de la cámara.



Figura 8. Cámara web. [5]



Figura 9. Raspberry Pi. [6]

- Un sistema basado en una ESP32 cam como la de la figura 10. Este es uno de los módulos más baratos para visión artificial del mercado, su precio ronda los 10 € y permite realizar tareas sencillas. Sin embargo, la reducida capacidad de cómputo de un ESP32 comparada con otros sistemas, así como el complejo desarrollo del programa necesario y la escasa fiabilidad que proporciona han sido motivos para descartar esta opción.



Figura 10. ESP32 Cam. [7]

- Cámara especializada. Hoy en día existen diferentes modelos en el mercado de cámaras de visión artificial que incluyen su propio microcontrolador. Estas cámaras están optimizadas para su uso en este tipo de proyectos e incluyen algoritmos especializados para la detección de formas, colores, caras etc....

A continuación, se muestran los tres principales modelos que se han considerado para este proyecto:

- OpenMV Cam RT1062 (figura 11):

Este modelo ronda los 110 €.

Se programa fácilmente con MicroPython, haciendo que su uso sea accesible tanto para principiantes como para profesionales. Lo que la hace especial es su enfoque específico en la visión artificial. Su principal ventaja son las grandes posibilidades de personalización que ofrece ya que es programada directamente mediante código, permitiendo al usuario crear el algoritmo más apropiado para la aplicación que vaya a realizar.

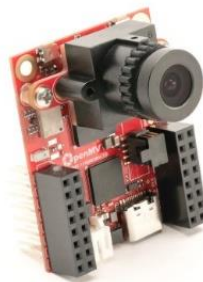


Figura 11. OpenMV Cam. [8]

- HuskyLens (figura 12):

Este cámara ronda los 55 € es la opción más barata entre sus semejantes. Este modelo incluye numerosas opciones como reconocimiento facial y aprendizaje mediante inteligencia artificial, así como una pantalla para su configuración.



Figura 12. HuskyLens. [9]

- Pixy Cam 2 (figura 13):

Este modelo ronda los 75 €. Cuenta con un espacio amigable y sencillo para su configuración y con diferentes algoritmos especializados para tareas como detección de colores, detección de etiquetas, seguimiento de líneas etc. No está preparada para el reconocimiento facial pero cuenta con una buena precisión, leds integrados para su iluminación en la oscuridad y está preparada para ser conectada a fácilmente a diferentes microcontroladores con diferentes protocolos de comunicación.



Figura 13. Pixy cam 2.1. [10]

SELECCIÓN DE COMPONENTES

Finalmente. La Pixy Cam fue la opción escogida. Las posibilidades de utilizar una Raspberry Pi o similar requerían un trabajo añadido de programación, así como una mayor cantidad de elementos por un precio similar, por lo que fueron descartadas. La ESP32 cam era una opción viable y barata pero la calidad de su cámara, su reducida velocidad de cómputo y la escasa fiabilidad de un software propio implementado en este módulo, llevaron a desestimar esta opción.

De entre los modelos de cámara con procesador integrado, la OpenMV Cam era la opción más costosa y sin algoritmos especializados para las aplicaciones que se requieren, mientras que la HuskyLens a pesar de ser más barata es de propósito más generalista y no cuenta con tanta precisión como la Pixy Cam.

Finalmente, el modelo de la Pixy Cam 2 ofrece una opción intermedia en precio, con buena precisión, algoritmos específicos de detección de colores y líneas, una comunidad y documentación que la respaldan, una interfaz amigable que permite realizar cambios en la configuración rápidamente y sin demasiados conocimientos y una fácil integración con microcontroladores como Arduino y ESP32.

Otra de las ventajas de esta cámara es que incluye la posibilidad de leer etiquetas específicas como se muestra en la figura 14. De esta forma, se pueden utilizar para la navegación del robot haciendo que se detenga, que reduzca o aumente su velocidad, que cambie a modo manual, que modifique su régimen de giro etc.

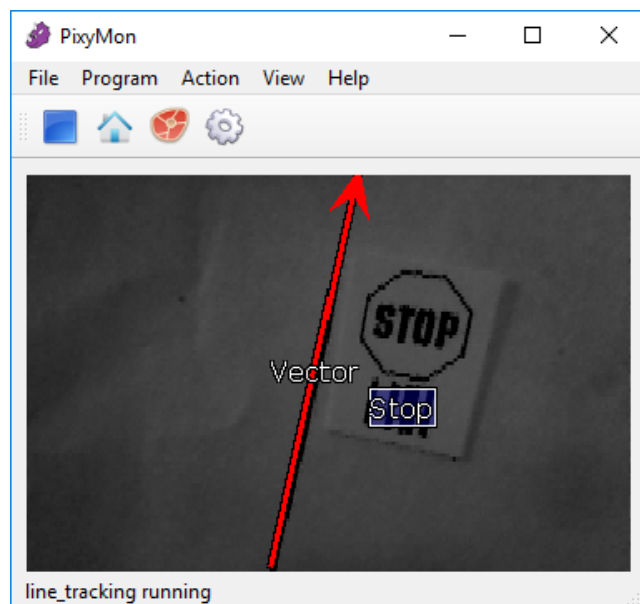


Figura 14. Lectura de una etiqueta. [11]

La Pixy cam 2 cuenta con un puerto micro USB B para ser conectada a un ordenador o alimentada. Además, cuenta con una serie de pines en la parte trasera que le permiten utilizar diferentes protocolos de comunicación como I2C, SPI, UART, ICSP etc. Podemos ver un esquema de los pines en la figura 15.

También dispone de salidas analógicas preparadas para conectar directamente servos, lo cual le permitiría enfocar de forma automática los objetos buscados.

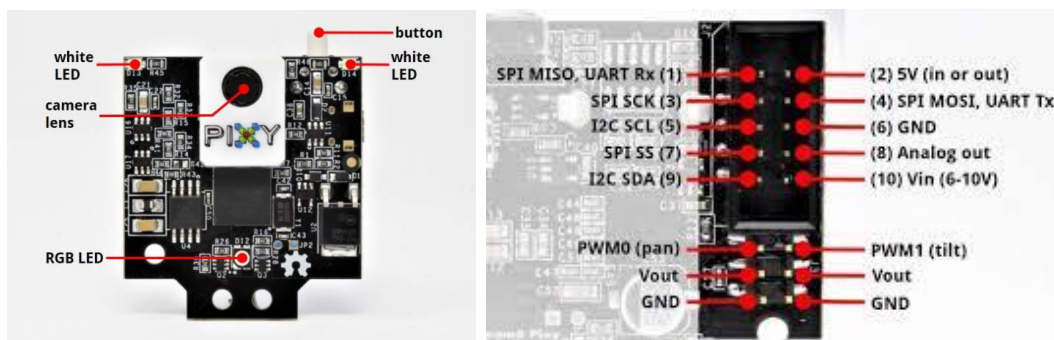


Figura 15. Pines de la Pixy cam 2. [12]

3.2 Selección del microcontrolador

Un microcontrolador es un pequeño circuito integrado que combina una CPU, memoria y puertos de entrada/salida en un solo chip. En este caso será el encargado de recibir las instrucciones de la cámara, funcionar como receptor de las órdenes del mando de control y de comunicarse con las placas de *hoverboard*.

Existen diferentes alternativas en el mercado, Arduino, ESP32, Teensy, Raspberry... Con diferentes modelos cada uno. En este caso se ha escogido un módulo ESP32, cuenta con el precio más económico y una gran variedad de funcionalidades (figura 16).

A diferencia de las placas de Arduino los ESP32 incluyen de forma integrada módulos bluetooth, wifi e incluso radio que podrán ser de gran utilidad para las funciones inalámbricas del robot. Además, su gran número de pines, su elevada capacidad de procesamiento y sus dos núcleos lo hacen tremendamente versátil.



Figura 16. ESP32 wroom. [13]

3.3 Selección de las placas y los motores

Para este proyecto se requiere una máquina capaz de desplazar con facilidad y seguridad un peso elevado.

Los *hoverboard* están diseñados para soportar y transportar el peso de una persona, lo que garantiza que sus motores y controladores de motor (ESC) sean capaces de manejar cargas significativas con eficiencia.

Además, estos componentes son fáciles de obtener y relativamente económicos debido a su producción en masa (figura 17). Utilizar motores, ruedas y baterías de *hoverboard* no solo reduce los costos, sino que también simplifica el diseño y la integración del AGV, proporcionando una solución práctica y accesible para proyectos de automatización y transporte de cargas moderadas.

Un solo *hoverboard* contiene:

- Dos motores brushless de gran potencia con sensores Hall para detectar la posición
- Una placa controladora que actúa de ESC (electronic speed control)
- Una batería de 36 V-42.5 V, de 4400 mAh y de hasta 10 A con cargador
- Dos giroscopios (en este caso no serán utilizados)



Figura 17. Motores y placas de un hoverboard. [14]

Los *hoverboard* tienen un rango de precios variable, pero uno económico ronda los 50 €.

La necesidad de garantizar un funcionamiento preciso y seguro de la máquina hacen como requisito la utilización de al menos 4 motores brushless para su construcción, por lo tanto, serán necesarias dos placas de control.

En cuanto a la alimentación, inicialmente se hará uso de dos de las baterías de 36V como las de la figura 18. Lo cual ofrece una autonomía más que suficiente, sin embargo, se explorarán otras opciones de alimentación en el apartado de líneas de mejora.

Estas baterías están pensadas para un *hoverboard* de entre 1 y 2h de autonomía el cual debe desplazar el peso de una persona, esto las hace apropiadas para esta aplicación ya que aportan una duración y potencia suficientes.



Figura 18. Batería de un hoverboard. [15]

3.4 Electrónica auxiliar

Como en cualquier circuito electrónico, la utilización de elementos auxiliares es necesaria, en este proyecto se han utilizado los siguientes componentes:

- Seta de emergencia IDEC YW1B-V4E02R (figura 19):
Su doble canal lo hace perfecto para cortar la alimentación de forma simultánea para las dos baterías utilizadas.



Figura 19. Seta de emergencia [16]

- Regulador de tensión L7805cv (figura 20):
Permite convertir tensiones de hasta 35 V en 5 V. Es utilizado para aprovechar una de las salidas de 15 V de las placas para alimentar la electrónica de control.

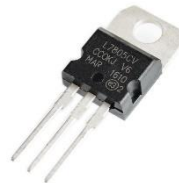


Figura 20. L7805. [17]

- Level shifter (figura 21):
Circuito electrónico que ajusta el nivel de voltaje de una señal de entrada para que sea compatible con el nivel de voltaje requerido por un circuito de salida diferente. En este caso nos permitirá convertir la señal de 3.3 V del ESP32 a los 5 V a los que funcionan las placas del *hoverboard*.

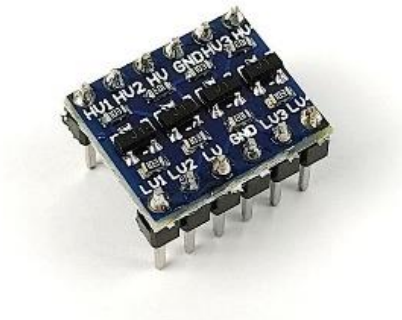


Figura 21. Level shifter. [18]

- Placas de circuito impreso perforadas (figura 22):
Componente típico que nos permitirá realizar los esquemas de conexiones de forma limpia.

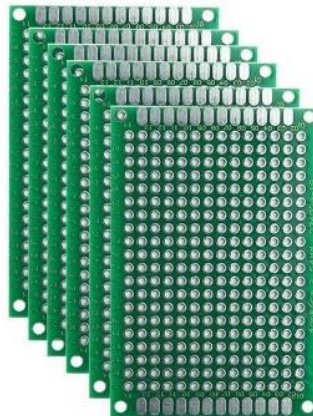


Figura 22. Placas perforadas. [19]

CAPITULO 4 PRIMEROS PASOS Y CONSTRUCCIÓN

4.1 Prototipado

Antes de comenzar con la construcción del modelo definitivo del robot era preciso comprobar el funcionamiento de la navegación mediante visión artificial que se ha planteado. Con ese objetivo, se ha diseñado y construido el pequeño prototipo de la figura 23 que imita la forma y propiedades del diseño del AGV real.

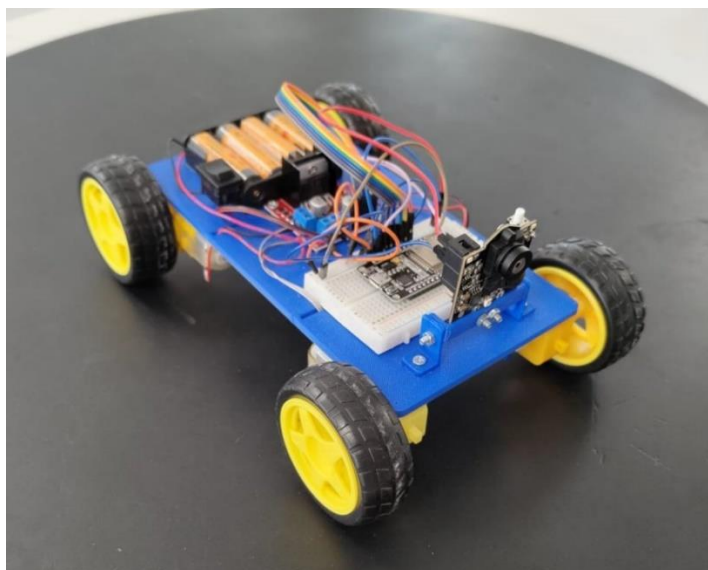


Figura 23. Prototipo del AGV

En su construcción se han empleado materiales sencillos y de bajo coste como una alimentación a base de pilas, un controlador de motores de corriente continua L298N, un ESP32 como el que se utilizará en la máquina y motores de 5 V imitando la colocación de los brushless del modelo final.

Las pilas alimentan en paralelo la cámara y el L298N que a su vez alimenta el ESP32 a través de su salida de 5 V y los motores.

Los pines del controlador están conectados al ESP32 el cual gobierna los motores siguiendo las indicaciones de la cámara tal y como se indica en el apartado del sistema de guiado.

Las tierras de todos los elementos del circuito están referenciadas a un mismo punto para garantizar una correcta alimentación y comunicación.

El esquema eléctrico del prototipo se encuentra en el anexo C del proyecto.

4.1.1 Diseño e impresión 3D

Para el chasis del prototipo se han diseñado e impreso en 3D algunas piezas con PLA.

Este material es un polímero biodegradable derivado del ácido láctico. Es un material altamente versátil, que se hace a partir de recursos renovables al 100%, como son el maíz, la remolacha, el trigo y otros productos ricos en almidón. Este ácido tiene características equivalentes a las del PETG (el plástico de las botellas de agua).

Para la realización del diseño se ha utilizado el programa FUSION 360 y una impresora Ender 3 Pro.

Es necesario asegurar el correcto diseño y ensamblaje de todos los componentes, por ello, se han utilizado los correspondientes modelos 3D de la electrónica utilizada.

En las figuras 24, 25, 26 y 27 podemos ver las vistas del diseño 3D desarrollado.

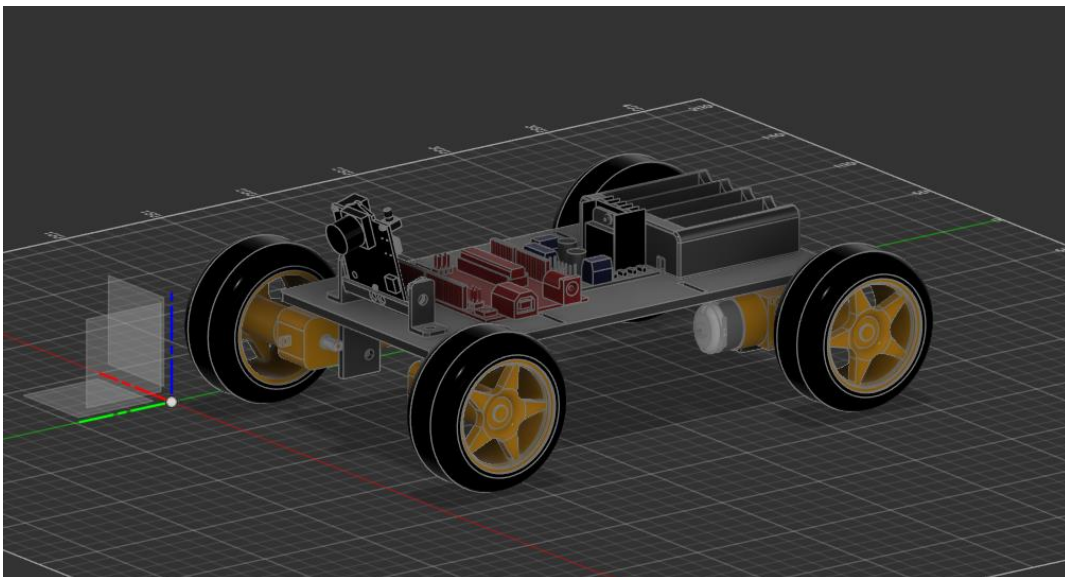


Figura 24. Perspectiva del prototipo

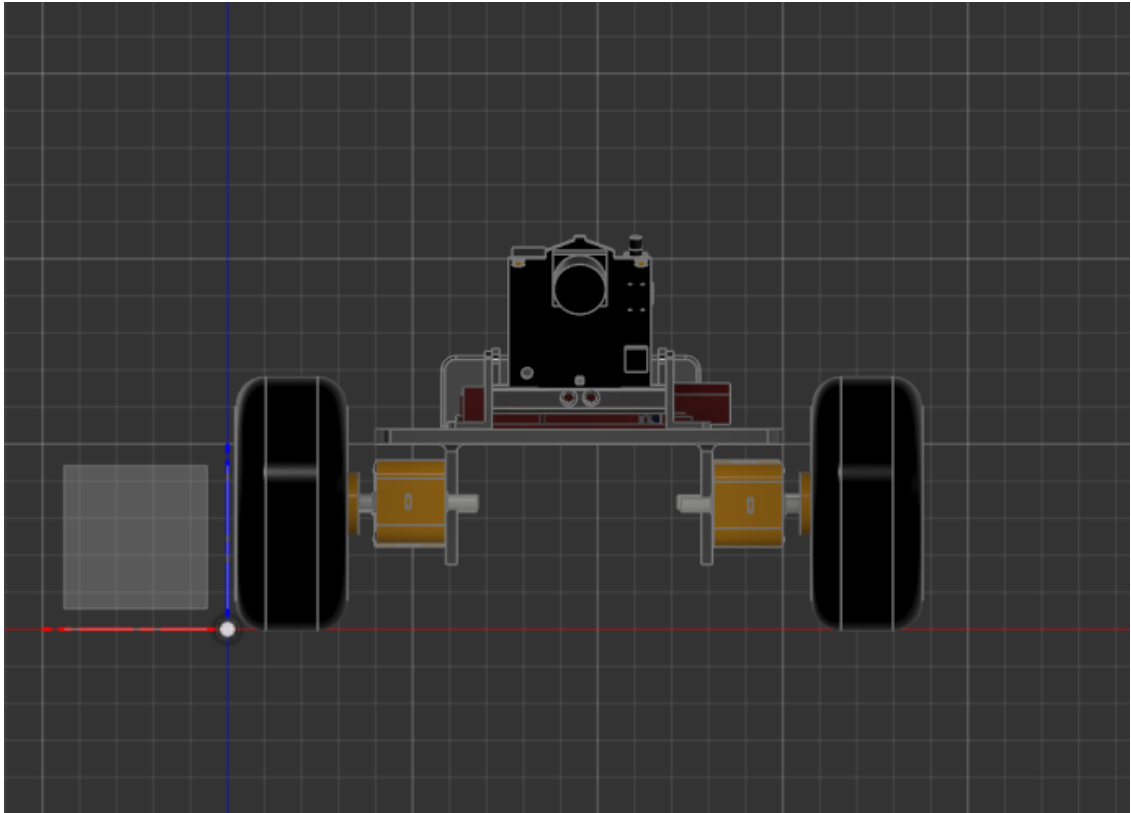


Figura 25. Alzado del prototipo

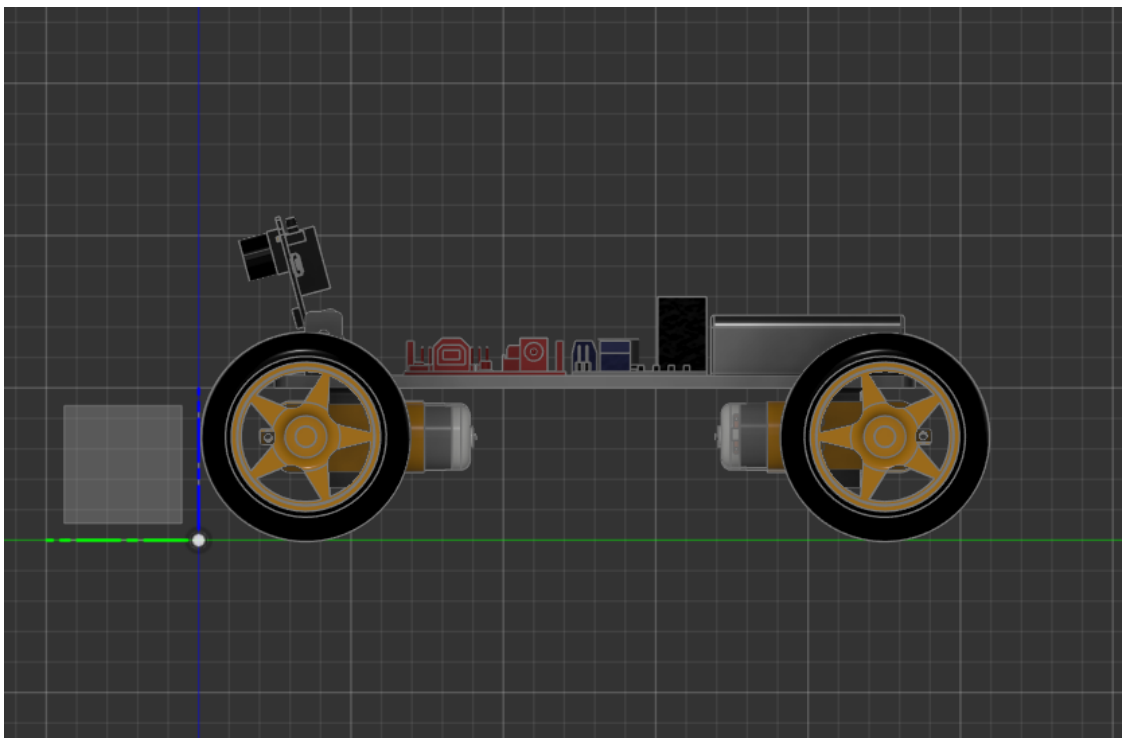


Figura 26. Perfil del prototipo

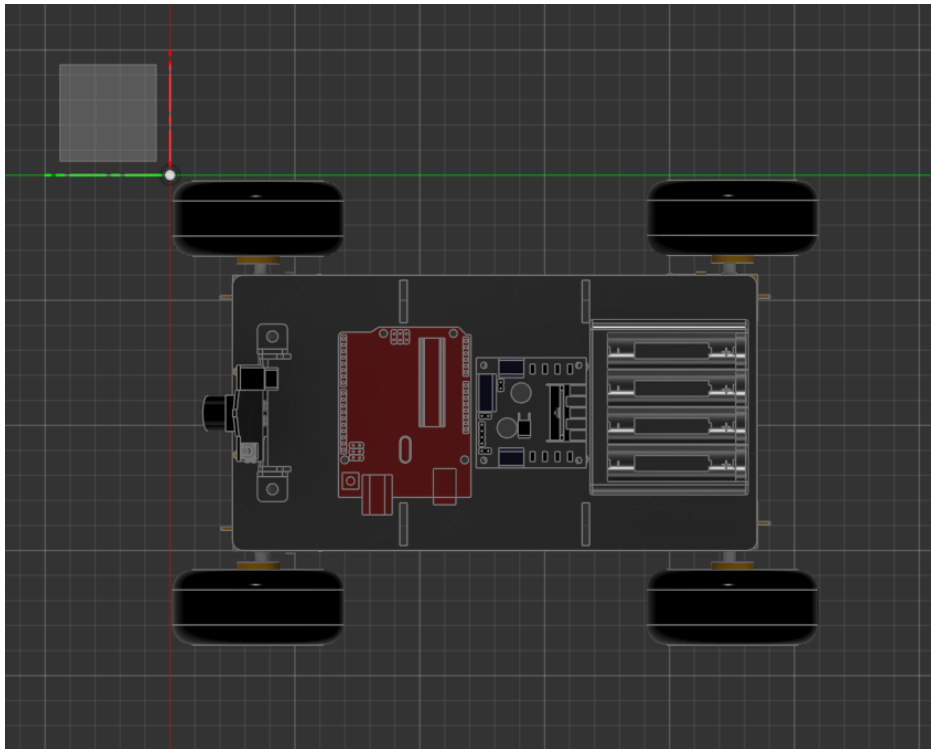


Figura 27. Planta del prototipo

4.1.2 Sistema de guiado

El propósito original del sistema de navegación es la detección y seguimiento de un cable diferenciable del entorno. Para ello lo primero que se pensó es la utilización del algoritmo especializado en seguimiento de líneas de la Pixy cam, al cual una vez ajustado se le aplicaría un filtro para que únicamente detectase las líneas del color deseado. En la figura 28 podemos ver cómo funciona este algoritmo.

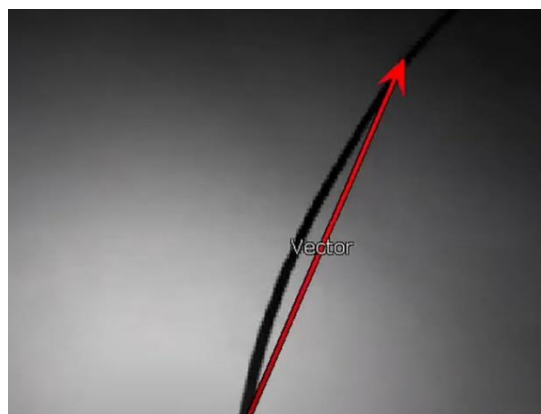


Figura 28. Pixy cam reconociendo líneas. [20]

Este sistema, si bien lograba detectar las líneas de forma correcta no era lo bastante preciso.

Deformaciones en el cable, cambios de anchura y especialmente, cambios en la luminosidad y contraste con el suelo podían confundir al algoritmo impidiendo un correcto funcionamiento.

Como segunda opción, se planteó la posibilidad de utilizar el algoritmo de reconocimiento de colores exclusivamente. Si el color del cable era lo bastante vistoso no sería confundido con otros elementos y su configuración era más sencilla que la del algoritmo de seguimiento de líneas.

Este sistema funcionó de forma muy satisfactoria, siendo más sencillo, más preciso, más fácil de configurar y más robusto que el anterior. Su comportamiento ante cambios de luminosidad era satisfactorio y era sencillo cambiar el color de detección si se requería.

Este algoritmo detecta un color específico dentro del espacio y lo marca con un bloque de las mismas dimensiones que el objeto detectado (figura 29). La cámara, mediante el protocolo de comunicación especificado, envía al microcontrolador las coordenadas del centro de dicho bloque dentro de su espacio de visión.

Finalmente, para realizar el seguimiento del cable, el microcontrolador sólo debe utilizar la coordenada horizontal para mandar la salida correspondiente a los motores en función de si el bloque se encuentra más o menos alejado del centro de visión de la cámara. Es decir, si el objeto se encuentra a la derecha del centro de visión el robot girará a la derecha, mientras que si se encuentra a la izquierda girará a la izquierda.

La programación del seguimiento de la referencia para el prototipo es simple. El robot gira hacia el sentido especificado deteniendo las ruedas de ese lado, es decir, para girar a la izquierda avanzará únicamente con las ruedas de la derecha hasta centrarse sobre el cable.

Esto es así ya que sólo se diseñó para familiarizarse con la cámara y comprobar su utilidad en esta aplicación concreta.

En el apartado de programación de la cámara se hablará de la implementación realizada en el AGV final para el seguimiento del cable y los tipos de control explorados (P, PD, PID).



Figura 29. AGV reconociendo el cable

En un principio, el prototipo se diseñó para albergar un Arduino UNO, ya que la cámara Pixy2 está preparada para una comunicación directa con esa placa mediante la comunicación ICSP. Posteriormente se cambió por un ESP32 para establecer la comunicación que será utilizada en la última fase del robot, tal y como se planeó en la selección de componentes.

Dado que los ESP32 carecen de ICSP, hubo que cambiar la forma de comunicación a I2C en la configuración de la cámara. Para comunicarse mediante I2C con un ESP32 es necesario utilizar la librería “wire” mediante la cual podemos seleccionar dos pines del ESP32 para que actúen como SDA (serial data) y SCL (serial clock).

Una vez preparado el ESP32 se conectó a la cámara para comenzar a recibir datos de las imágenes tomadas. Los pines designados como SDA y SCL se conectaron a los pines 9 y 5 de la cámara respectivamente. El protocolo de comunicación de la cámara puede seleccionarse desde las opciones de su programa de control “PixyMon”.

4.2 Chasis del AGV

El chasis está formado por perfiles metálicos de 6 cm x 3 cm formando un rectángulo de 47 cm x 35.5 cm. Como base para la electrónica y tapa para soportar el *shaker* se han empleado tablas de 49.5 cm x 35 cm con 1.5 cm de espesor.

Los perfiles cuentan con agujeros que permiten la introducción de los cables de control y de los cables de los motores. Además, se han colocado en un lateral los dos puertos de carga de las baterías como se muestra en la figura 30.



Figura 30. Puertos de carga

Como método de anclaje de los motores al chasis se han empleado piezas y tornillos originales del *hoverboard* a los que se les ha diseñado un adaptador que ha sido impreso en 3D, mostrado en la figura 31.

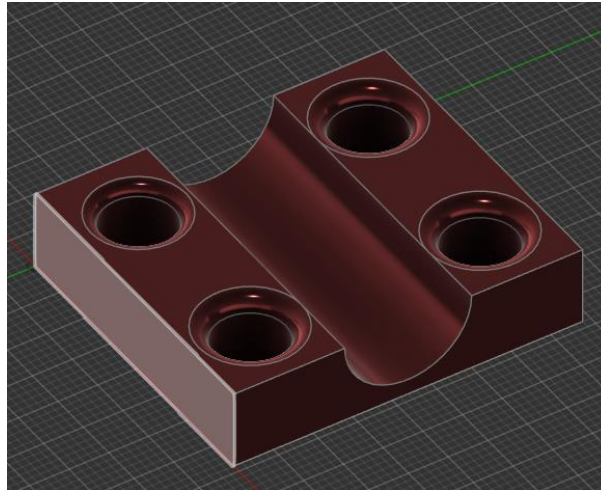


Figura 31. Adaptador de los motores

Esta pieza fue originalmente diseñada e impresa en el trabajo previo, sin embargo, con el uso y la presión de los tornillos de fueron deformando y deteriorando (figura 32). Esto es debido a que no fueron impresas con el suficiente relleno para la aplicación que se deseaba.

Dado que los archivos originales se han perdido, se han rediseñado unas piezas nuevas tomando las antiguas de referencia y se han impreso con una densidad del 100% para evitar nuevas deformaciones.



Figura 32. Adaptador antiguo

Salvo el rediseño de los adaptadores, el resto de las piezas fueron reutilizadas todas del trabajo previo del cual se ha mantenido la mayor parte del diseño.

4.3 Sistemas de seguridad

Esta máquina cuenta con numerosos riesgos para las personas. Se trata de un vehículo que puede llegar a transportar cargas de alrededor de 100 Kg, con centro de gravedad bajo, que puede alcanzar velocidades de hasta 30 km/h con grandes aceleraciones y con baterías inflamables.

Por ello es necesario colocar una serie de medidas de seguridad que garanticen la integridad del AGV, de su carga y de las personas que se encuentren alrededor.

4.3.1 Toma de tierra al chasis

Conectar la toma de tierra de la alimentación del robot al chasis es una práctica recomendable en sistemas eléctricos y electrónicos de vehículos y otras máquinas.

Sirve como protección ante cortocircuitos, facilitando que ante este caso la corriente vuelva a través del chasis en lugar de atravesar a una posible persona. Por lo tanto, esto es otro sistema de seguridad, tanto para la integridad del equipo como para las de las personas. Además, permite absorber las cargas electrostáticas que se generan con su uso y actúa como protección ante emisiones electromagnéticas que puedan causar interferencias en nuestros sistemas de comunicación.

Para realizar esta conexión, se ha aprovechado uno de los agujeros de los perfiles metálicos que componen el chasis para introducir un tornillo que aprisione con seguridad el cable de tierra como se ve en la figura 33.

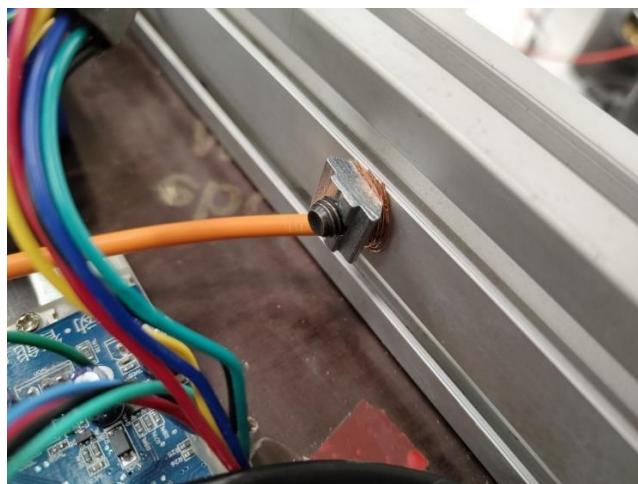


Figura 33. Toma de tierra

4.3.2 Limitación de la velocidad por software

Como se explicará posteriormente en el apartado de reprogramación de las placas, se han realizado modificaciones en la configuración de los controladores de los motores.

Estas modificaciones tienen como objetivo asegurar que ningún fallo de programación, señal o interferencia vayan a afectar a la velocidad del vehículo convirtiéndolo en un peligro.

Para ello se ha limitado la velocidad de los motores a un máximo de 50 rpm que equivaldría aproximadamente a 1.5 km/h de velocidad máxima.

4.3.3 Seta de emergencia e interruptor

Originalmente tanto los *hoverboard* como el diseño previo del AGV carecen de medidas para cortar la alimentación. Esto no sólo supone un desgaste de las baterías con el tiempo, también implica la incapacidad de cortar la alimentación ante un cortocircuito.

Por ello se ha instalado una doble medida de seguridad, un interruptor y una seta de emergencia en serie.

El interruptor nos permite encender y apagar el robot con mayor facilidad (sin tener que utilizar los dos botones originales). Además, evitamos la descarga de las baterías y reducimos el peligro de fallo eléctrico mientras el robot no está en funcionamiento.

La seta de emergencia permite disponer de un botón grande y fácilmente alcanzable que permitirá cortar la alimentación de la máquina en caso de peligro. Esto evitará accidentes como descontrol del vehículo, enganche en alguna de las ruedas, ignición de una batería etc.

4.4 Integración de la electrónica

Una vez preparado el chasis con las modificaciones necesarias y ubicados los elementos de seguridad se procede a la colocación de la electrónica.

El esquema eléctrico completo del AGV se encuentra en el anexo C del proyecto. En él se muestra la integración completa de todos los elementos relevantes del robot, a falta de los motores y sensores hall cuya descripción no aporta nada relevante.

4.4.1 Electrónica de potencia

La electrónica de potencia está conformada por las baterías, los controladores de los motores y los elementos de seguridad.

La colocación de las placas se mantiene según el diseño del trabajo previo. Se colocan ambas con la misma orientación atornilladas directamente a la tabla que ejerce de base.

4.4.2 Electrónica de control

Para la instalación de la cámara de visión artificial y del ESP32 en el cuerpo del robot se ha diseñado y fabricado una pieza en 3D que hace de soporte a la electrónica (figura 35).

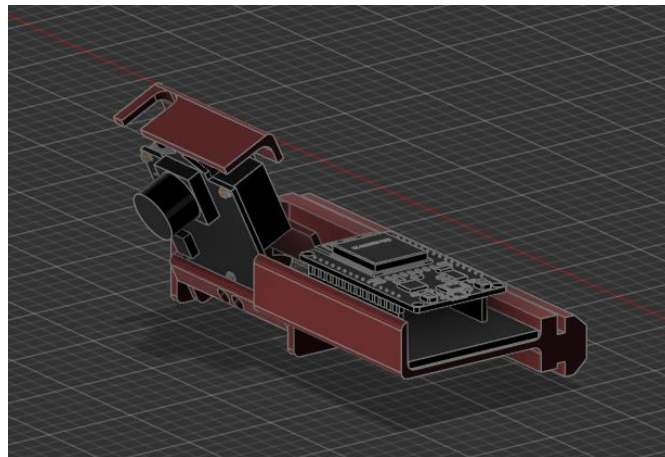


Figura 35. Soporte para la electrónica de control

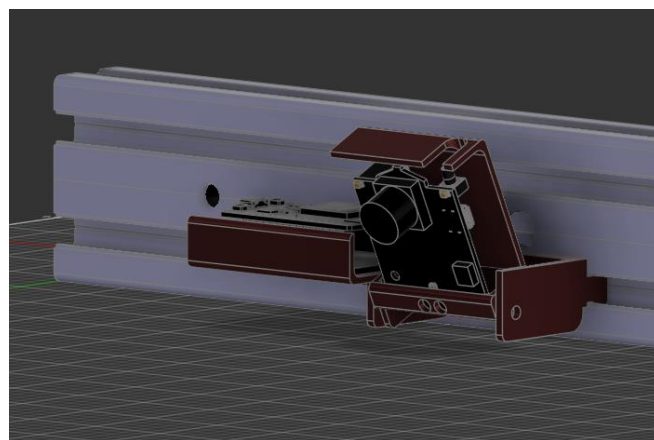


Figura 36. Soporte para la electrónica de control encajado en el perfil

Como se puede ver en la figura 36, esta pieza está diseñada para encajar perfectamente en el perfil de forma que no es necesario ningún tipo de sujeción adicional. A través de uno de los agujeros del perfil pasan todos los cables relacionados con el control del robot, tanto los de comunicación del ESP32 con las placas como los de alimentación de la electrónica.

El ESP32 se ha instalado en un zócalo soldado a una placa perforada en la que se han realizado las conexiones correspondientes para el circuito de comunicación con el robot.

La cámara y el ESP32 reciben una alimentación de 5 V en paralelo del interior del robot (ver electrónica auxiliar).

Todas las decisiones son gestionadas desde el ESP32 el cual es el único que se comunica con los controladores de los motores. La cámara por otro lado únicamente se comunica con el ESP32 informándole de los objetos que detecta y de su posición en el espacio.

4.4.3 Electrónica auxiliar

La electrónica auxiliar en este proyecto tiene dos cometidos.

Por un lado, rebajar la salida de 15 V de las placas para alimentar la electrónica de control. Por otro lado, adecuar las señales emitidas por el ESP32 para asegurar la comunicación con las placas.

Para realizar la alimentación de la electrónica de control se utilizarán las salidas de 15 V de ambas placas en paralelo que serán rebajadas mediante dos reguladores de tensión L7805 igualmente en paralelo.

La utilización de ambas salidas de 15 V y no sólo de una está relacionada con el consumo de la electrónica de control, el cual es demasiado elevado para una sola de las placas, hay más información sobre esto en el apartado de problemas encontrados y soluciones.

Respecto a la utilización de dos L7805 en paralelo en lugar de uno se debe al calentamiento excesivo al que están sometidos, por lo tanto, son utilizadas dos unidades para compartir la corriente y por lo tanto la disipación de calor (de nuevo, ver el apartado de problemas encontrados y soluciones).

Los L7805 son componentes con tres pines de los cuales el central corresponde a GND mientras que los otros dos corresponden a una entrada de hasta 35 V y a una salida de 5 V regulados. Pueden soportar hasta 1A de corriente.

Estos componentes han sido soldados a una de las placas perforadas donde se ha realizado el punto de unión de todas las tierras del circuito. De su salida de 5 V parte un cable que se encarga de la alimentación de la electrónica de control y otro que sirve de voltaje de referencia para el level shifter.

Este es un componente encargado de convertir señales de 3.3 V a 5 V y viceversa. Su aplicación en este proyecto es necesaria ya que el controlador elegido (ESP32) funciona con lógica a 3.3 V mientras que las placas de control de los motores funcionan a 5 V.

Esto hace indispensable cambiar la tensión de las señales del microcontrolador para adecuarlas a este nuevo rango de tensiones.

El level shifter además de recibir las señales de entrada y de salida, requiere una referencia de voltaje de ambos niveles, es decir, una referencia de 3.3 V para el nivel bajo y de 5 V para el nivel alto.

Los L7805 se encargan de dar esta referencia de 5 V además de la alimentación mientras que la salida de 3.3 V del ESP32 se encarga de la referencia del nivel bajo.

CAPITULO 5 CONTROL Y PROGRAMACIÓN

Una vez realizada la construcción mecánica y la instalación de todos los componentes electrónicos con las comprobaciones necesarias para saber que se ha realizado correctamente, el robot está listo para ser configurado y programado.

5.1 Reprogramación de las placas y control de los motores

Uno de los elementos clave del proyecto es la reconversión de las placas de hoverboard como ESC personalizados. El método y los conocimientos necesarios para realizar esta tarea se describirán a lo largo de los siguientes apartados.

5.1.1 Características de las placas

Como se ha mencionado anteriormente, el *firmware* con el que se reprogramó las placas en el trabajo previo realizado en el robot se ha perdido, por lo tanto, es necesario reprogramarlas de nuevo.

Para ello, lo primero es investigar y conocer las características de la placa utilizada:

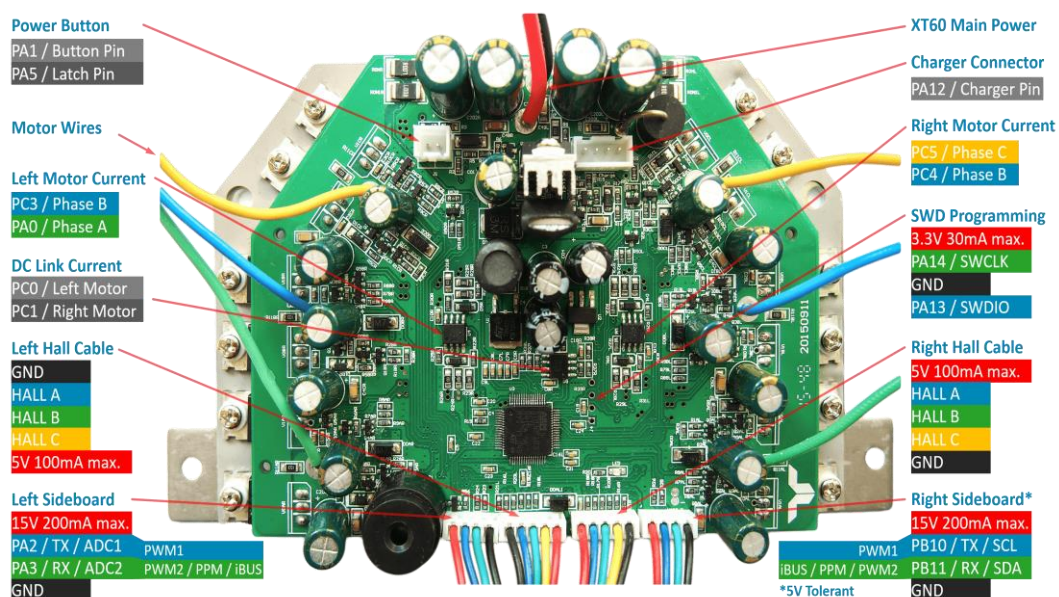


Figura 37. Esquema de la controladora. [21]

La PCB, mostrada en la figura 37, cuenta con numerosos elementos entre los que destacamos los siguientes:

- Cuatro pines de conexión con el STM32 que gobierna la placa, permitiendo su programación mediante SWD.
- Ocho pines de conexión con los giroscopios, mediante los cuales podremos realizar una comunicación con el ESP32. Además, ofrecen una salida 15 V que se ha aprovechado para la alimentación de la electrónica. Las cuatro conexiones de la izquierda funcionan a 3.3 V, mientras que las de la derecha toleran hasta 5 V.
- 10 pines de conexión con los sensores Hall que nos permiten conocer la posición de los motores.

5.1.2 Preparación de la placa y selección del *firmware*

Mediante los pines centrales, los cuales se corresponden con 3.3 V, GND, SWCLK y SWDIO podremos conectarnos al STM32 y reprogramarlo. Para ello es necesaria una herramienta electrónica como la de la figura 38, llamada ST-link.



Figura 38. ST-link. [22]

A través de este dispositivo, se pueden conectar los pines de la placa directamente al USB del ordenador donde gracias al software STM32CubeProgrammer, podremos reconfigurarla (figura 39). El primer paso es realizar un borrado completo del chip para prepararlo para cargar el nuevo *firmware*.

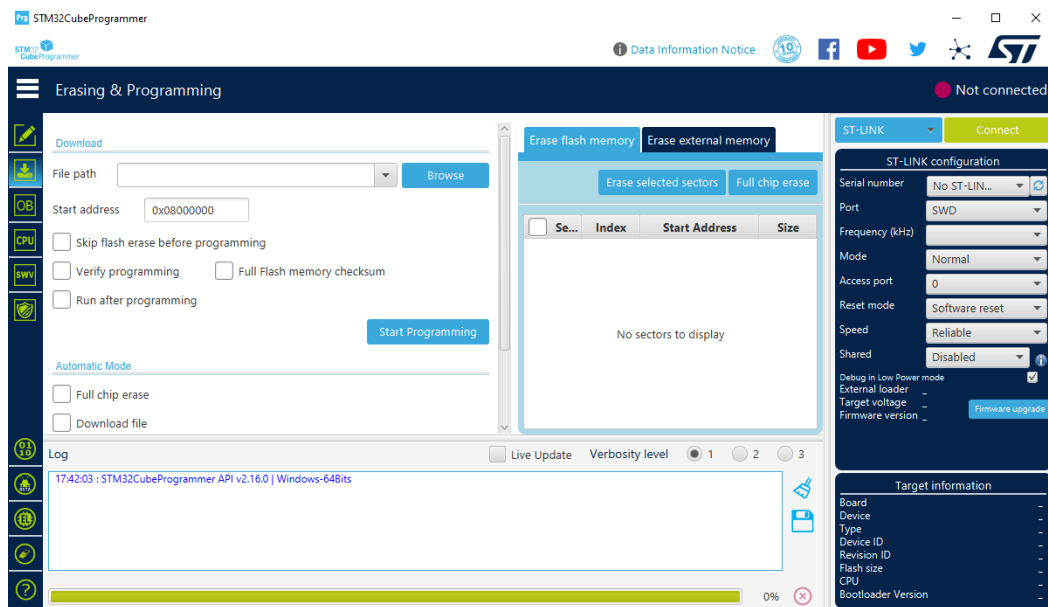


Figura 39. STM32CubeProgrammer

Existe variedad de *firmware* diferentes en internet para reprogramar este tipo de placas. Tras realizar una investigación de los disponibles en internet el que se ha seleccionado es un *firmware* que utiliza el control FOC para los motores.

FOC (Field-Oriented Control), o control orientado al campo en español, es una técnica avanzada utilizada en el control de motores brushless, como los motores de corriente continua sin escobillas (BLDC) y los motores síncronos de imanes permanentes (PMSM). Esta técnica permite un control muy preciso del par y la velocidad del motor al manipular las corrientes que pasan por las bobinas del motor de manera coordinada con el campo magnético del rotor.[23]

En lugar de tratar directamente con las corrientes trifásicas del motor, FOC convierte estas corrientes a un sistema de coordenadas que gira con el rotor del motor. Por lo general, utilizan controladores proporcional-integral (PI) donde los componentes actuales se comparan con valores de referencia, en lugar de utilizar modulación de ancho de pulso (PWM). Esto permite que los motores

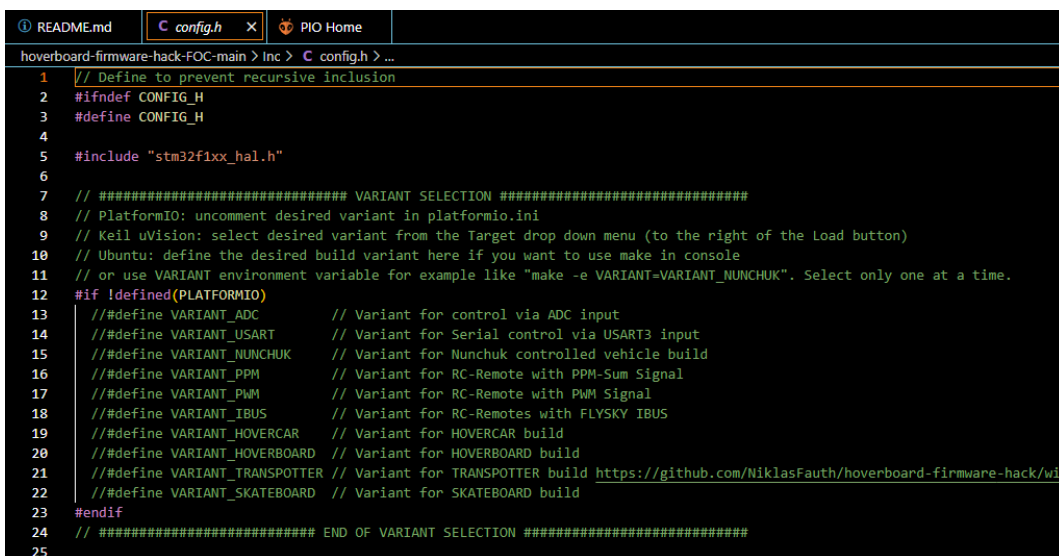
eléctricos funcionen suavemente en todo el rango de velocidades y generen el par máximo a velocidad cero.

Otro beneficio del control orientado al campo es que puede ofrecer una rápida aceleración y desaceleración del motor, brindando un control más preciso en motores de alto rendimiento.

FOC ofrece varias ventajas significativas. Optimiza el uso del flujo magnético y del par, lo que mejora la eficiencia energética del motor. Además, reduce el ruido y la vibración, proporcionando un control más suave y preciso. También mejora la capacidad de respuesta del motor ante cambios en la carga y la velocidad, lo que es especialmente útil en aplicaciones que requieren un control preciso y eficiente del motor, como en este proyecto donde se requiere que el movimiento sea lento, fluido y con un control efectivo del par.

5.1.3 Modificaciones del *firmware*

Este *firmware* no ha sido seleccionado únicamente por su control FOC, sino que además permite múltiples sistemas de comunicación con las placas mediante la modificación del archivo “config.h” (figura 40)



```
1 // Define to prevent recursive inclusion
2 #ifndef CONFIG_H
3 #define CONFIG_H
4
5 #include "stm32f1xx_hal.h"
6
7 // ##### VARIANT SELECTION #####
8 // PlatformIO: uncomment desired variant in platformio.ini
9 // Keil uVision: select desired variant from the Target drop down menu (to the right of the Load button)
10 // Ubuntu: define the desired build variant here if you want to use make in console
11 // or use VARIANT environment variable for example like "make -e VARIANT=VARIANT_NUNCHUK". Select only one at a time.
12 #if !defined(PLATFORMIO)
13 // #define VARIANT_ADC // Variant for control via ADC input
14 // #define VARIANT_USART // Variant for Serial control via USART3 input
15 // #define VARIANT_NUNCHUK // Variant for Nunchuk controlled vehicle build
16 // #define VARIANT_PPM // Variant for RC-Remote with PPM-Sum Signal
17 // #define VARIANT_PWM // Variant for RC-Remote with PWM Signal
18 // #define VARIANT_IBUS // Variant for RC-Remotes with FLYSKY IBUS
19 // #define VARIANT_HOVERCAR // Variant for HOVERCAR build
20 // #define VARIANT_HOVERBOARD // Variant for HOVERBOARD build
21 // #define VARIANT_TRANSPOTTER // Variant for TRANSPOTTER build https://github.com/NiklasFauth/hoverboard-firmware-hack/wiki
22 // #define VARIANT_SKATEBOARD // Variant for SKATEBOARD build
23 #endif
24 // ##### END OF VARIANT SELECTION #####
25
```

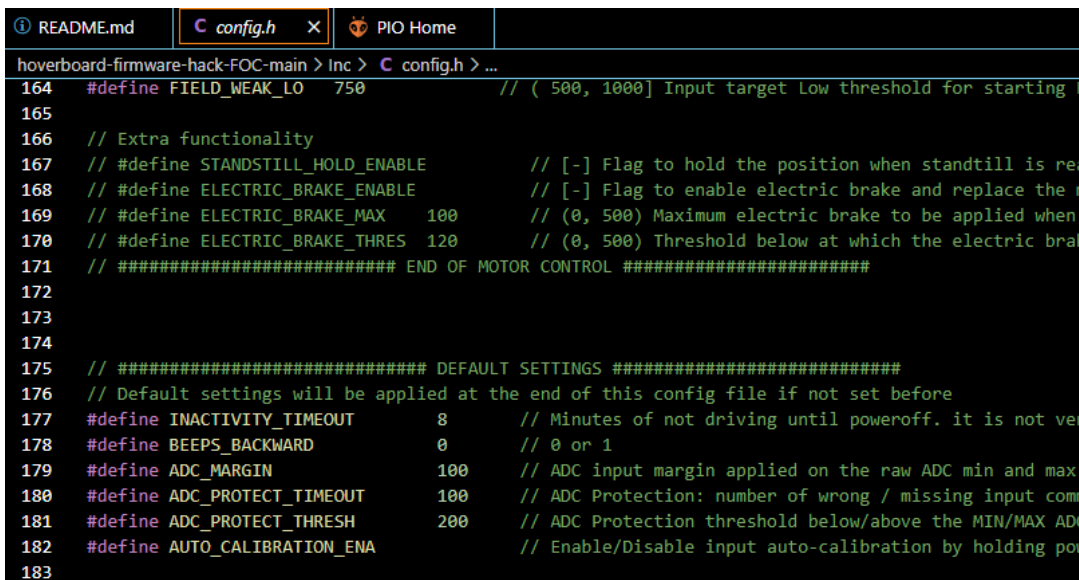
Figura 40. Archivo config.h

De esta forma, tan sólo descomentando la opción deseada podemos acceder a sistemas de comunicación basados en señales analógicas (para ser controlado por potenciómetros o joysticks), USART, I2C, PPM, PWM o IBUS.

Además del sistema de comunicación, tenemos la posibilidad de personalizar más aspectos del funcionamiento de las placas.

CONTROL Y PROGRAMACIÓN

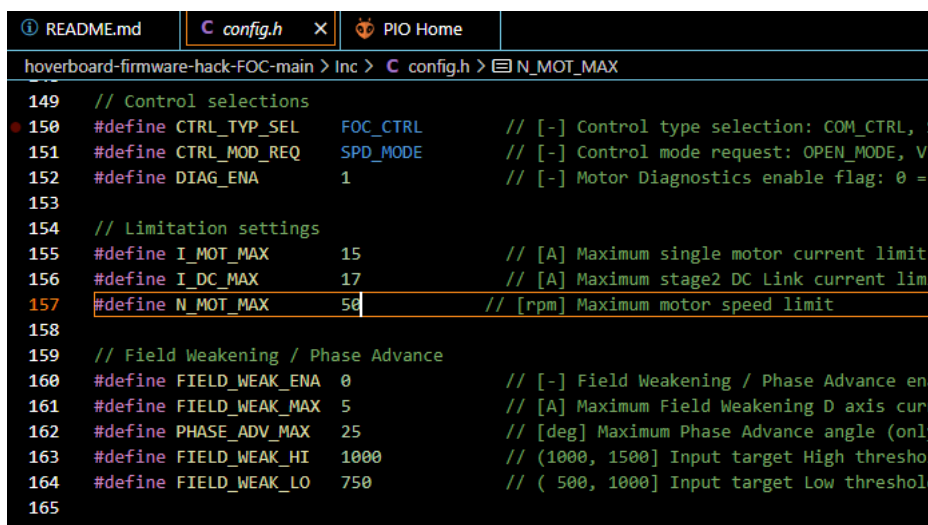
Originalmente, los *hoverboard* realizan un pitido como medida de seguridad cuando circulan marcha atrás de forma similar a como lo haría un vehículo de obra. Para esta aplicación este pitido es irrelevante y molesto por lo que se ha decidido eliminarlo modificando de nuevo el archivo “config.h” mediante la sustitución del número 1 por un 0 en la línea 178 (figura 41).



```
164 #define FIELD_WEAK_LO 750 // ( 500, 1000] Input target Low threshold for starting
165
166 // Extra functionality
167 // #define STANDSTILL_HOLD_ENABLE // [-] Flag to hold the position when standstill is re
168 // #define ELECTRIC_BRAKE_ENABLE // [-] Flag to enable electric brake and replace the r
169 // #define ELECTRIC_BRAKE_MAX 100 // (0, 500) Maximum electric brake to be applied when
170 // #define ELECTRIC_BRAKE_THRES 120 // (0, 500) Threshold below at which the electric bra
171 // ##### END OF MOTOR CONTROL #####
172
173
174
175 // ##### DEFAULT SETTINGS #####
176 // Default settings will be applied at the end of this config file if not set before
177 #define INACTIVITY_TIMEOUT 8 // Minutes of not driving until poweroff. it is not ve
178 #define BEEPS_BACKWARD 0 // 0 or 1
179 #define ADC_MARGIN 100 // ADC input margin applied on the raw ADC min and max
180 #define ADC_PROTECT_TIMEOUT 100 // ADC Protection: number of wrong / missing input com
181 #define ADC_PROTECT_THRESH 200 // ADC Protection threshold below/above the MIN/MAX AD
182 #define AUTO_CALIBRATION_ENA // Enable/Disable input auto-calibration by holding po
183
```

Figura 41. Código de conjuración del zumbador

Como medida de seguridad ante posibles fallos lógicos o de comunicación con el ESP32, se ha decidido limitar la velocidad del vehículo (figura 42). Esto es posible hacerlo también mediante la modificación de la configuración en el *firmware*:



```
149 // Control selections
150 #define CTRL_TYP_SEL FOC_CTRL // [-] Control type selection: COM_CTRL, S
151 #define CTRL_MOD_REQ SPD_MODE // [-] Control mode request: OPEN_MODE, VL
152 #define DIAG_ENA 1 // [-] Motor Diagnostics enable flag: 0 =
153
154 // Limitation settings
155 #define I_MOT_MAX 15 // [A] Maximum single motor current limit
156 #define I_DC_MAX 17 // [A] Maximum stage2 DC Link current limit
157 #define N_MOT_MAX 50 // [rpm] Maximum motor speed limit
158
159 // Field Weakening / Phase Advance
160 #define FIELD_WEAK_ENA 0 // [-] Field Weakening / Phase Advance ena
161 #define FIELD_WEAK_MAX 5 // [A] Maximum Field Weakening D axis curr
162 #define PHASE_ADV_MAX 25 // [deg] Maximum Phase Advance angle (only
163 #define FIELD_WEAK_HI 1000 // (1000, 1500] Input target High threshol
164 #define FIELD_WEAK_LO 750 // ( 500, 1000] Input target Low threshol
165
166 // Extra functionality
```

Figura 42. Limitación máxima de la velocidad

En la línea 157, puede especificarse la velocidad máxima de los motores en revoluciones por minuto. Estos motores pueden alcanzar grandes velocidades sin embargo para esta aplicación se considera que con una velocidad máxima de 50 rpm es suficiente, lo que equivaldría aproximadamente a 1.5 km/h.

Para terminar, para lograr el objetivo de un movimiento lento, pero con un par motor elevado es necesario realizar un cambio más en el programa. Esto supuso un problema importante cuando en un inicio, se lograba un control satisfactorio de la velocidad y dirección de los motores, pero se observó una carencia importante de par a velocidades bajas. Posteriormente se descubrió la existencia de tres modos de funcionamiento basados cada uno en la potencia, la velocidad y el par motor, cuya selección se realiza en la línea 151.

Cada uno de ellos mantiene una de estas tres variables constante siguiendo las indicaciones del usuario, por lo tanto, seleccionando el modo de velocidad constante obtenemos como resultado que el motor tratará de mantener la referencia de velocidad indicada realizando los cambios que sean necesarios. Es decir, si el controlador detecta que la rueda se mueve más despacio que lo requerido aumentará el par hasta alcanzar la referencia. De esta manera se logró un par elevado a bajas revoluciones.

5.1.4 Sistemas de comunicación

Originalmente, se planteó la posibilidad de utilizar la comunicación USART para poder enviar señales desde el ESP32 ya que dispone de esta posibilidad. Sin embargo, esta comunicación no pudo realizarse correctamente.

El principal inconveniente de estas placas es que no siempre son iguales entre sí, pequeñas modificaciones del hardware entre modelos pueden crear problemas de comunicación e incompatibilidad con ciertas funciones del *firmware*. Aparentemente la comunicación vía USART no está disponible para las placas de las que se dispone y por tanto, es necesario buscar una alternativa.

La segunda alternativa fue la comunicación vía I2C, este sistema está preparado para funcionar a través de un mando de videoconsola. Primero se comprobó la disponibilidad del sistema a la comunicación I2C utilizando el mando original para el que está preparado.

El resultado fue satisfactorio, se logró mover de forma efectiva todos los motores en diferentes sentidos.

El siguiente paso fue lograr replicar el funcionamiento del mando mediante el ESP32 para suplantarlos y enviar las órdenes a través de él. Esto, sin embargo, no fue posible.

Tras varios intentos infructuosos tratando de replicar la dirección y comandos del mando se decidió que era mejor probar otro sistema diferente por lo que se trató de realizar la comunicación mediante señales analógicas.

Este sistema no admite referencias de voltaje generadas mediante modulación de ancho de pulso, por lo tanto, no sería posible crear estas señales directamente a través del ESP32. Al intentarlo los motores se movían de manera brusca y aleatoria. Para solucionarlo se planteó la posibilidad de utilizar un convertidor de señal analógica.

La idea consistía en utilizar unos dispositivos electrónicos como los de la figura 43, que permiten convertir una señal de PWM en una señal analógica.

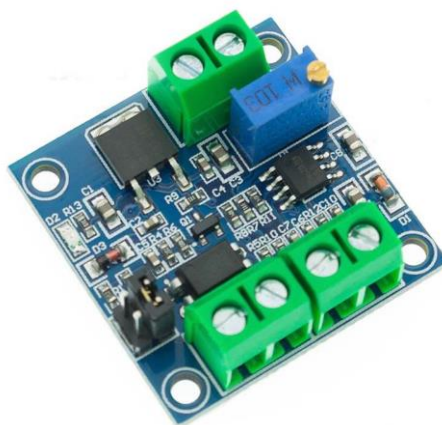


Figura 43. Convertidor de PWM a señales analógicas. [24]

No obstante, antes de comenzar con el diseño del circuito y su instalación, se observó que este sistema de control originalmente no está pensado para permitir que el vehículo retroceda. La posibilidad del robot de dar marcha atrás a la hora de maniobrar es algo indispensable y por lo tanto esta idea también fue descartada.

Finalmente, el sistema elegido que ha sido utilizado en el resto del proyecto es el de la modulación por ancho de pulso (PWM).

Originalmente esta comunicación está pensada para realizarse entre un receptor de radio y las placas, sin embargo, mediante la utilización de una librería para controlar servomotores, las señales del ESP32 son leídas correctamente controlando plenamente la velocidad y dirección de los motores.

La librería ESP32Servo.h ha sido la utilizada para realizar el control de los motores. Si bien originalmente está pensada para servos, permite emular la señal que emitiría un módulo de radio y por lo tanto permite la comunicación con la placa del robot. No sería válido utilizar una señal PWM tradicional con la función "analog.write()" sin haber sido configurada antes de forma apropiada ya que las placas no están preparadas para comunicarse mediante la modulación de pulso estándar que realiza el ESP32.

Para este sistema, es necesario conectar los cables Rx y Tx de cada placa a salidas del ESP32. Las señales emitidas a través de los cables Tx controlarán la velocidad del vehículo mientras que las emitidas a través del Rx controlarán la dirección.

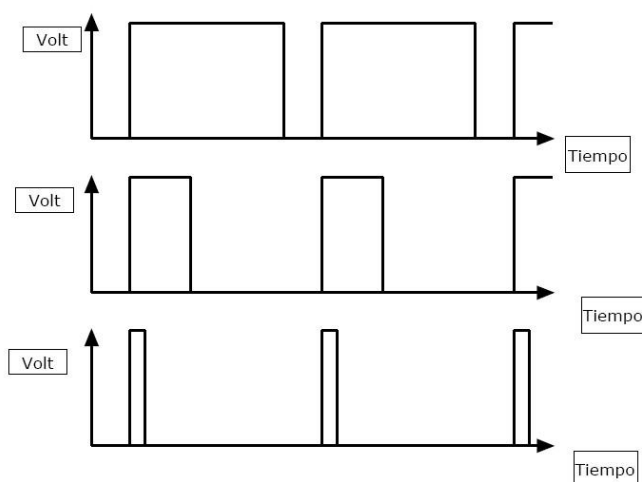
En la disposición original de las placas una controlaba los motores delanteros y otra los traseros, sin embargo, tal y como se explica en el apartado seis, esto se ha modificado. Actualmente una placa controla los dos motores derechos mientras que la otra los motores izquierdos.

Con esta configuración podemos controlar la velocidad y dirección del vehículo mediante sólo dos cables por los que se transmite una señal PWM. A través de uno de ellos se indicará el sentido y velocidad de los motores del lado derecho y a través del otro los del lado izquierdo.

De esta manera el robot podrá girar en ambos sentidos elevando o reduciendo la velocidad de los motores de un lado respecto de los del otro

El PWM es una técnica que consiste en variar el tiempo que una señal eléctrica está encendida o apagada manteniendo su voltaje y frecuencias constantes. En la figura 44 podemos ver un ejemplo de la modulación de la señal.

Figura 44. Esquema de funcionamiento del PWM. [25]



De esta forma podemos obtener valores eficaces de voltaje en el rango de tensión del que se disponga. También puede ser usado como medio de comunicación mediante la lectura de tiempos de la parte alta de la señal.[26]

En esta comunicación es necesario cumplir una serie de requisitos para que las señales sean correctamente interpretadas. Hay que emplear una frecuencia de 50 Hz, es decir, un periodo de 20 ms y la parte alta de la señal debe variar entre 1 ms y 2 ms. En un servo, un valor de 1 ms equivaldría a la posición más baja (0°) y 2 ms a la posición más alta (180°) como se puede ver en la figura 45. Sin embargo, en este caso una duración de 1 ms corresponderá a un 100% de la velocidad en sentido negativo, una duración de 1.5 ms corresponderá a la detención de los motores y una duración de 2 ms corresponderá a un 100% de la velocidad en sentido positivo. Los valores intermedios corresponden a valores intermedios de velocidad en ambos sentidos.[27]

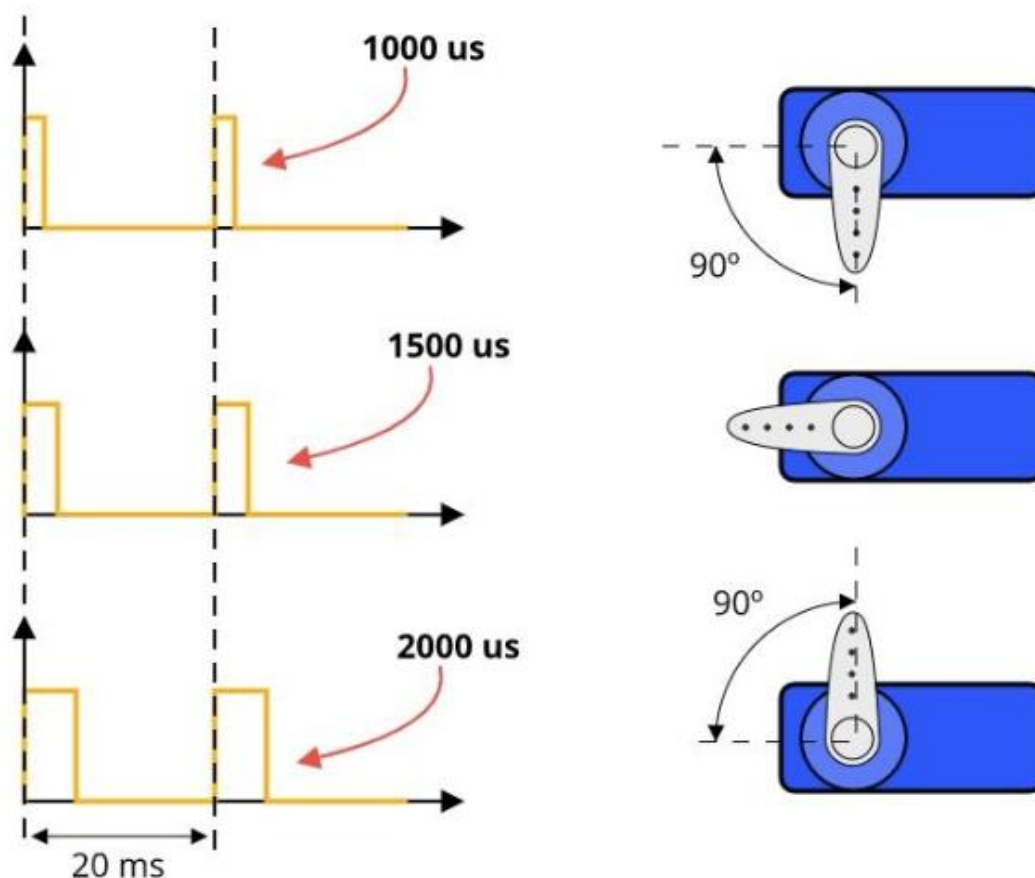


Figura 45. Esquema de funcionamiento de los servos. [28]

En cuanto al control de la dirección se hará de la misma manera. En este caso 1 ms corresponde a un giro total a la izquierda y 2 ms a un giro total a la derecha. Al igual que antes 1.5 ms equivaldría a centrar la dirección y el resto de los valores intermedios a giros más o menos pronunciados en ambos sentidos.

5.1.5 Programación del movimiento

En el código, para configurar la velocidad y dirección de los motores se crea un objeto “Servo” para cada señal, dos direcciones y dos velocidades (aunque con esta configuración sólo usaremos dos señales) y se les asigna un pin en la función “setup()” de nuestro código (figura 46).

```
Servo velocidad_trasera; //Señal PWM que controla el acelerador
Servo velocidad_delantera; //Señal PWM que controla el acelerador

Servo direccion_trasera; //Señal PWM que controla la dirección (volante)
Servo direccion_delantera; //Señal PWM que controla la dirección (volante)

bool bandera_movimiento = false; //Variable que sirve para que la rampa de

void configurarDispositivos() {
    // Configuraciones de dispositivos
    velocidad_delantera.attach(18);
    direccion_delantera.attach(5);

    direccion_trasera.attach(21);
    velocidad_trasera.attach(19);
    Serial.begin(9600);
}
```

Figura 46. Código de configuración

Luego para establecer la velocidad y dirección sólo es necesario utilizar la función “writeMicroseconds ()” para establecer el valor deseado (figura 47).

```
void detenerse() {
    bandera_movimiento=false; //Ponemos a 0 la bandera para indicar que el AGV está detenido

    velocidad_trasera.writeMicroseconds(1500); //Ponemos la velocidad a 0 (1500 equivale a no moverse)
    velocidad_delantera.writeMicroseconds(1500);

    direccion_trasera.writeMicroseconds(1500); //Fijamos la dirección a 0 (1500 equivale a no girar, si no tratará de pivotar)
    direccion_delantera.writeMicroseconds(1500);
    delay(10);
}
```

Figura 47. Función para detener el robot

5.2 Programación del modo automático

El modo automático del AGV es una de las piezas clave del proyecto.

Para cumplir con las necesidades planteadas y operar con seguridad debe cumplir los siguientes requisitos:

- Una vez encendida la máquina este modo sólo se activará por orden del operario.
- La máquina no comenzará su movimiento si no detecta el cable que debe seguir y deberá detenerse tan pronto como deje de verlo.
- El modo automático debe poder ser desactivado en cualquier momento por parte de los operarios
- La velocidad inicial del modo automático debe ser la mínima operativa del vehículo.
- La velocidad de desplazamiento debe poder ser manipulada en cualquier momento durante el trayecto del AGV.

Para cumplir estos requisitos se propone el siguiente planteamiento del programa.

El algoritmo que debe ejecutar el robot durante el modo automático se define en la función "void loop()" del código. Esto significa que será ejecutada cíclicamente mientras el robot se mantenga encendido.

Para evitar su activación durante el modo manual, el modo automático se encuentra dentro dos condiciones, la desactivación del modo manual por parte del operario y la detección del cable por parte de la cámara.

De esta forma, si el operario no ordena comenzar el modo automático o la cámara no ve claramente un objetivo que debe seguir el robot no comenzará a moverse.

El algoritmo ejecutado se basa en la detección del cable por parte de la cámara, la cual transmite al ESP32 mediante I2C la posición del cable dentro de su espacio de visión. El microcontrolador extrae el valor de la posición horizontal del cable y lo compara con la posición central de la cámara. De esta manera se puede obtener el error de la posición del robot.

Con este sencillo cálculo, se puede ajustar la velocidad de los motores de ambos lados del vehículo para que realicen giros para seguir la ruta de forma adecuada.

Para ajustar estos giros se ha implementado un controlador PID

Un controlador o regulador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una acción Proporcional, Integral y Derivativa. En la figura 48 podemos ver un sistema de lazo cerrado regido por un PID.[29]

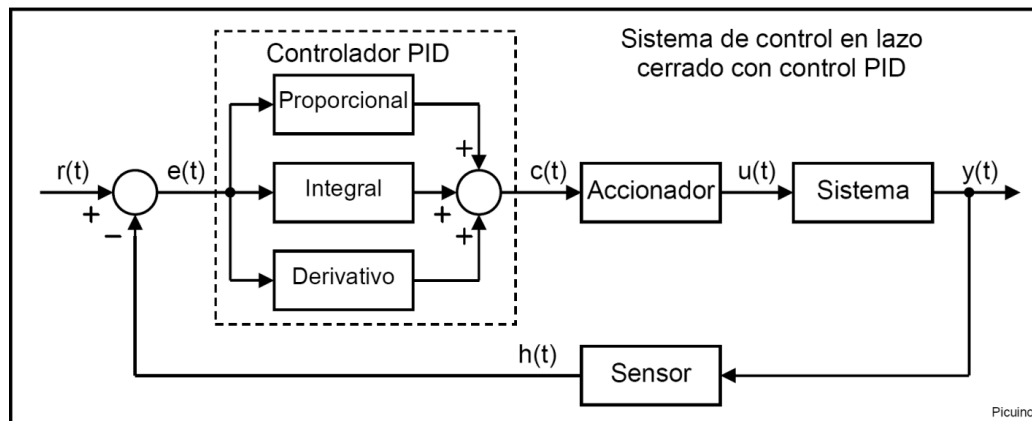


Figura 48. Sistema de lazo cerrado con PID. [30]

En nuestro caso la referencia sería la posición central de la cámara y el error la diferencia entra la referencia y el valor horizontal del cable transmitido por la cámara. El accionador serían los motores y el sistema la posición del robot.

Mediante este método podemos controlar y ajustar perfectamente la reacción de los motores ante una entrada dada y hacer que siga la ruta de forma fluida.

Para ello se ha programado un controlador PID que mediante el ajuste de los parámetros k_p , k_i y k_d controla el impacto de sus componentes en la salida.

Las características de estos componentes son las siguientes:

-Proporcional: Es la respuesta al error que se tiene que entregar de manera inmediata, es decir, si nos encontramos en el centro de la línea, los motores, tendrán en respuesta una velocidad de igual valor, si nos alejamos del centro, uno de los motores reducirá su velocidad y el otro aumentará.

-Integral: La integral es la sumatoria de los errores acumulados, tiene como propósito el disminuir y eliminar el error en estado estacionario provocado por el modo proporcional, en otras palabras, si el AGV se encuentra mucho tiempo alejado del centro, la acción integral se ira acumulando e irá disminuyendo el error hasta llegar al punto central.

-Derivativo: Es la derivada del error, su función es mantener el error al mínimo, corrigiéndolo proporcionalmente con la misma velocidad que se produce, de esta manera evita que el error se incremente, en otra palabra, anticipara la acción evitando así las oscilaciones excesivas.

La salida del controlador será la descrita según la ecuación 1.

$$Salida = P * kp + I * Ki + D * Kd$$

Ecuación 1

De forma que controlando el valor de estas constantes se controla su impacto en la salida. Estos parámetros deben ajustarse en función de las características del robot y las condiciones del terreno en el que opere.

La manipulación de las constantes tiene efectos diversos en el comportamiento del robot:

Factor (Kp) - Si el valor es excesivo, el robot tendera responder inestablemente, oscilando excesivamente. Si el valor es muy pequeño, el robot responderá muy lentamente, tendiendo a salirse de las curvas

Factor (Ki) - El valor excesivo de este provocara oscilaciones excesivas, Un valor demasiado bajo no causara impacto alguno.

Factor (Kd) - Un valor excesivo provocara una sobre amortiguación, que a su vez provoca inestabilidad.

Dado que el movimiento del robot varía con la carga y con el terreno, lo mejor es hacer un ajuste del PID en condiciones similares a las de campo. Es decir, el controlador se ajustará hasta seguir correctamente la línea, pero no se hará el ajuste final hasta no disponer del *shaker* para que su comportamiento en un entorno real sea el mejor posible.

Esto no implica que se requiera de una calibración cada vez que se cambie de carga y de terreno, sin embargo, conviene hacerlo en unas condiciones similares a las del uso habitual del vehículo.

5.3 Mando de control

Uno de los requisitos de este proyecto es la realización de un modo manual que permita maniobrar el robot por un operario. Este sistema de control debe ser inalámbrico ya que el operario no va a acompañar al robot en su recorrido. Está pensado para desplegar y recoger el robot en ambos extremos de su ruta y no para realizarla manualmente.

Dadas estas circunstancias, en esta primera versión del robot, es innecesario que la comunicación tenga un gran alcance y por lo tanto se ha descartado un sistema de radio. En el apartado de mejoras se hablará más sobre esta posibilidad.

Otras posibles opciones son la utilización de una red wifi o una comunicación bluetooth. Este último sistema ofrece una configuración más sencilla y robusta ante interferencias lo que lo hace idóneo para el proyecto.

Se ha planteado como mando inicial una aplicación móvil gratuita llamada "Bluefruit Conect" la cual permite conectarnos a dispositivos bluetooth para enviar una serie de ordenes por diferentes métodos.

En esta ocasión interesa el modo "control pad", que permite usar un mando con cuatro botones de dirección y cuatro botones auxiliares como se ve en la figura 49.



Figura 49. Mando de control

Este mando, es sencillo pero funcional y cumplirá con los requisitos para establecer un modo manual que permita maniobrar la máquina a distancias cortas y medias.

Para su programación se han creado dos librerías que simplifican el código y facilitan su comprensión y manipulación, cuyos nombres son “bluetooth.h” y “motores.h”

La librería “bluetooth.h” contiene las instrucciones relacionadas con la comunicación entre el robot y el mando, almacena los valores recibidos a través de la comunicación bluetooth y asigna la función correspondiente a cada botón.

En la librería “motores.h” se encuentran la configuración y las funciones desarrolladas para el control de los motores. Entre ellas se encuentran las necesarias para avanzar, retroceder, girar, detenerse etc.

También se han incluido mecanismos para que cada vez que el robot se desplace en cualquier sentido se realice una rampa de aceleración que impide cambios bruscos en la velocidad. De esta manera se evita que la carga pueda caer por inercia, se mejora la maniobrabilidad y se fuerzan menos los motores.

Los botones auxiliares “1”, “2”, “3” y “4” son utilizados para el cambio entre el modo manual y el modo automático y para realizar cambios en la velocidad del vehículo.

El botón “1” cambia el modo de manual a automático y viceversa. Como es lógico el robot siempre comienza en modo manual para evitar accidentes.

Los botones “2”, “3” y “4” corresponden a las tres velocidades a las que puede desplazarse el robot, siendo estas la mínima velocidad a la que el robot se mueve con fluidez (2), la intermedia la cual duplica la velocidad mínima (3) y la máxima velocidad la cual es el triple que la mínima (4).

Como se ha comentado anteriormente, por motivos de seguridad se ha modificado el *firmware* de las placas para establecer una velocidad máxima de 50 rpm, con estas ruedas equivale aproximadamente a algo menos de 1.5 km/h. Por tanto, las velocidades media y mínima serán de aproximadamente 1 km/h y 0.5 km/h respectivamente.

CAPITULO 6 PROBLEMAS ENCONTRADOS Y SOLUCIONES

Durante la realización del TFG han surgido multitud de problemas e imprevistos que han tenido que ser investigados y corregidos. Algunos de ellos han obligado a realizar modificaciones en el diseño.

6.1 Interferencias electromagnéticas

Una vez finalizada la construcción del robot se procedió al desarrollo de la programación, para ello se situó el robot en un lugar elevado con las ruedas suspendidas en el aire.

Durante la programación se llevaron a cabo pruebas del modo manual con resultados satisfactorios, sin embargo, aparecieron problemas cuando se colocó el robot en el suelo.

Al tratar de controlar el robot en un entorno real y con carga (su propio peso) este comenzó a mostrar inestabilidad y movimientos erráticos. Los motores se movían en ráfagas cortas en ambos sentidos y con aleatoriedad de forma que resultaban incontrolables. A pesar de eso el robot mostraba signos de recibir las órdenes mandadas desde el mando.

Para tratar de resolver este problema se comprobó nuevamente el circuito, el cableado, la carga de las baterías, la alimentación etc. Pero no se localizó ningún error.

Un fallo de programación fue descartado rápidamente ya que cuando el robot se colocaba suspendido en el aire respondía perfectamente a las órdenes del mando.

Fue entonces cuando se planteó la posibilidad de interferencias electromagnéticas. Esta hipótesis sugería que al someter los motores a una carga como es lógico aumentaba el consumo de corriente y que ésta producía interferencias en las señales de control. Esto explicaría por qué al eliminar la carga el robot funcionaba correctamente. Sin carga el consumo era mucho menor y las interferencias electromagnéticas también.

Para probar esta hipótesis se realizó una prueba para tratar de aislar las señales de control. El método utilizado fue envolver los cables en papel de aluminio y añadir una capa exterior de cinta aislante que previniese de posibles cortocircuitos.

El papel de aluminio, al ser conductor de la electricidad puede actuar como jaula de Faraday, bloqueando campos eléctricos externos y creando una zona libre de cargas en el interior.

Una vez “apantallados” los cables de control se probó de nuevo el robot en el suelo y el resultado fue satisfactorio. Tal y como se planteó los campos electromagnéticos generados por los motores provocaban interferencias en las señales de control del robot.

Finalmente, una vez localizado el problema, la solución provisional de utilizar papel de aluminio de cocina fue sustituida por cinta adhesiva de aluminio que ofrece una alternativa más duradera y efectiva, aunque lo ideal sería utilizar cable apantallado especializado lo cual se propone como una posible mejora del proyecto.

6.2 Alimentación de la electrónica de control

Durante la integración de la cámara como sistema de guiado del robot, se detectaron de nuevo problemas de inestabilidad y fallos en el control.

Estos fallos consistían principalmente en un movimiento confuso del robot que no respondía adecuadamente. En el modo manual el vehículo realizaba movimientos erráticos y aleatorios de forma similar al problema de interferencias electromagnéticas. En el modo automático se descontrolaba totalmente sin seguir la ruta marcada.

Durante la realización de algunas pruebas se observó un error en la secuencia de encendido de los LED de iluminación de los que dispone la cámara. En condiciones normales debería encender un LED morado y acto seguido uno blanco, sin embargo, esto no ocurría. Además, la luz de estos LED brillaba con menor intensidad.

Esto originó la hipótesis de la falta de potencia en la alimentación de la electrónica de control y descartó las sospechas de un fallo de programación.

Para confirmarlo se probó nuevamente la máquina desconectando la alimentación de la cámara y por lo tanto encendiendo sólo el microcontrolador. El resultado fue positivo, el robot respondía con normalidad.

En un principio se estimó que el consumo de la electrónica de control no debería ser demasiado elevado y que la salida de 15 V de uno de los controladores de los motores debería ser suficiente para alimentarla. Sin embargo, fue una suposición equivocada.

Para solucionarlo, se han conectado las salidas de 15 V de ambas placas en paralelo duplicando así la corriente máxima que pueden suministrar.

Una vez realizada la modificación en el circuito se probó de nuevo el robot con toda la electrónica alimentada y respondió adecuadamente en ambos modos de funcionamiento.

6.3 Sobrecalentamiento

Solucionado el problema de alimentación de la electrónica comenzaron los ajustes de la programación en condiciones reales.

Una vez pasado un tiempo de funcionamiento del robot, comenzaron a fallar algunas de sus funciones.

Se observó que los LED de la cámara que iluminan su campo de visión comenzaron a parpadear. Además, el robot en el modo manual perdía alguna de las órdenes emitidas por el mando de control, comenzando a girar indefinidamente e ignorando la orden de detención, por ejemplo.

Dado que en un inicio el vehículo parecía funcionar correctamente, se comprobó si al reiniciarlo se solucionaba el problema. La respuesta fue que sí pero sólo durante unos breves instantes.

Al repetir esta prueba más veces se observó que el tiempo en el que el robot funcionaba correctamente aumentaba si se dejaba más tiempo apagado entre prueba y prueba.

Estos indicios apuntaban a un sobrecalentamiento y por lo tanto se realizó una inspección del circuito de alimentación tanto de la electrónica de control como de la electrónica de potencia. El resultado fue la localización de un sobrecalentamiento en el L7805, el regulador de tensión que proporciona 5 V al microcontrolador y a la cámara.

El consumo de la electrónica de control se ha estimado en alrededor de 500mAh mientras que el L7805 puede proporcionar hasta 1 A, por ello fue considerado apto para su utilización en el circuito. Sin embargo, este sobrecalentamiento afecta a su funcionamiento.

Se plantearon dos alternativas para solucionar este problema:

- La utilización de un convertidor Buck para sustituir al L7805. Este componente realiza una conversión del voltaje mucho más eficiente y sin calentamiento.
- La utilización de dos L7805 en paralelo.

La opción finalmente escogida fue esta última.

Utilizar un convertidor Buck sería la solución óptima, sin embargo, para instalarlo habría que hacer grandes modificaciones en el circuito de alimentación a riesgo de dañar algún otro componente o pista. Además, es un componente más costoso que habría que encargar.

Los L7805 son componentes funcionales, económicos y se dispone de ellos en el laboratorio. Su instalación es mínimamente invasiva y el resultado es similar por eso se ha optado por esta opción.

La colocación de dos L7805 en paralelo permite compartir la corriente haciendo que cada uno de ellos soporte la mitad de intensidad. Al estar sometidos a la mitad de corriente, los componentes se calentarán menos y entrarán dentro de su rango de funcionamiento.

Tras realizar la instalación se probó de nuevo a utilizar el vehículo durante periodos largos sin detectarse los fallos mencionados. También se comprobó como la temperatura de los reguladores de tensión había disminuido notablemente.

6.4 Control limitado de la dirección

En un principio, el cableado del AGV se había realizado de forma que una de las placas controlase los dos motores delanteros mientras que la otra controlaría los traseros. Este planteamiento se heredó del trabajo previo del robot y se mantuvo a lo largo del proyecto.

Sin embargo, una vez se quiso realizar un ajuste de los giros del vehículo para que funcionase de forma fluida, se detectó una limitación importante de este diseño.

Tal y como funciona el *firmware* actual de la máquina, cada placa puede realizar un control sobre la dirección de ambos motores por separado y de la velocidad de ambos. Sin embargo, no permite realizar un control de la velocidad de cada motor por separado.

Es decir, las opciones para girar son:

- Que las ruedas de un lado vayan en un sentido y las del otro vayan en el contrario
- Que las ruedas de un lado vayan en un sentido y las del otro se detengan

Esto imposibilita el hacer giros progresivos regulando la velocidad de las ruedas. Haciendo que las ruedas de la derecha vayan a máxima velocidad y las de la izquierda a una velocidad intermedia, por ejemplo.

A la hora de programar un control para que el robot siga de forma fluida la ruta supone un problema importante.

Para solucionar este problema se planteó la posibilidad de cambiar el cableado de los motores, de manera que una placa controle los dos motores de la derecha y la otra los dos motores de la izquierda.

Dado que los cables eran lo suficientemente largos, no fue necesario cambiar la ubicación de las placas dentro del chasis del robot.

Este cambio hizo necesaria una modificación en programación de las funciones de movimiento que se habían realizado para ahora controlar de forma independiente la velocidad de ambos lados del robot.

En un principio se hacía uso de los dos cables de comunicación de cada placa (Rx y Tx) donde uno nos permitía controlar el avance y retroceso de los motores y el otro si se pretendía que girasen en sentidos opuestos.

Actualmente, sólo es necesario uno de los cables de comunicación de cada placa. Esto se debe a que ya no pretendemos que los motores de una misma placa giren en sentidos opuestos. Con un solo cable controlamos la velocidad de ambos y su sentido.

Esta configuración nos permite controlar de forma independiente la velocidad y sentido de ambos lados del vehículo y por lo tanto ajustar los giros sin limitaciones.

CAPITULO 7 IMPLEMENTACIÓN Y RESULTADOS

Una vez finalizada la construcción y programación del AGV se realizaron una serie de pruebas para comprobar su alcance.

Primero se realizaron pruebas sin añadir peso extra, en las que se comprobó el correcto funcionamiento de los controles del modo manual, y el seguimiento adecuado de la ruta en el modo automático. Este último permitió realizar ajustes en aspectos como el ángulo de la cámara o la utilización o no de los LEDs de iluminación.

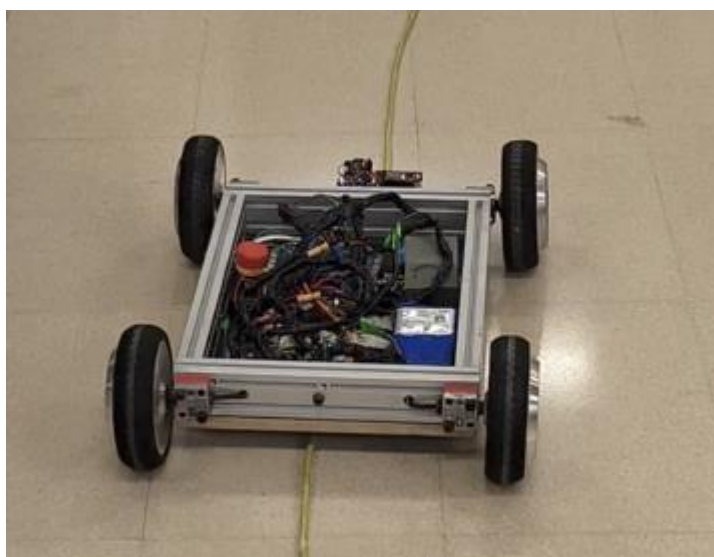


Figura 50. AGV siguiendo la ruta

Para el trazado de las rutas se utilizó un alargador de un color verde llamativo sobre el suelo de baldosas del laboratorio. En la figura 50 podemos ver el robot en funcionamiento.

Como resultado se confirmó un movimiento adecuado del vehículo, un buen funcionamiento de las rampas de aceleración y una detección adecuada de la línea en un suelo especialmente difícil ya que refleja mucho la luz y no es muy diferenciable del cable utilizado.

El control de la velocidad en el modo automático fue ajustado para un seguimiento más fluido de la ruta y se decidió realizar la calibración de la cámara siempre con el LED de iluminación. No sólo mejora el rendimiento eliminando brillos, también permite una buena independencia de la luz ambiente.

También se puso a prueba la capacidad del vehículo de realizar giros cerrados. Si bien su cometido es realizar rutas rectilíneas al estar trazadas con un cable presentarán irregularidades, por lo que es conveniente ajustar el seguimiento. Además, la realización de curvas relativamente cerradas permite la utilización del AGV en otros usos.

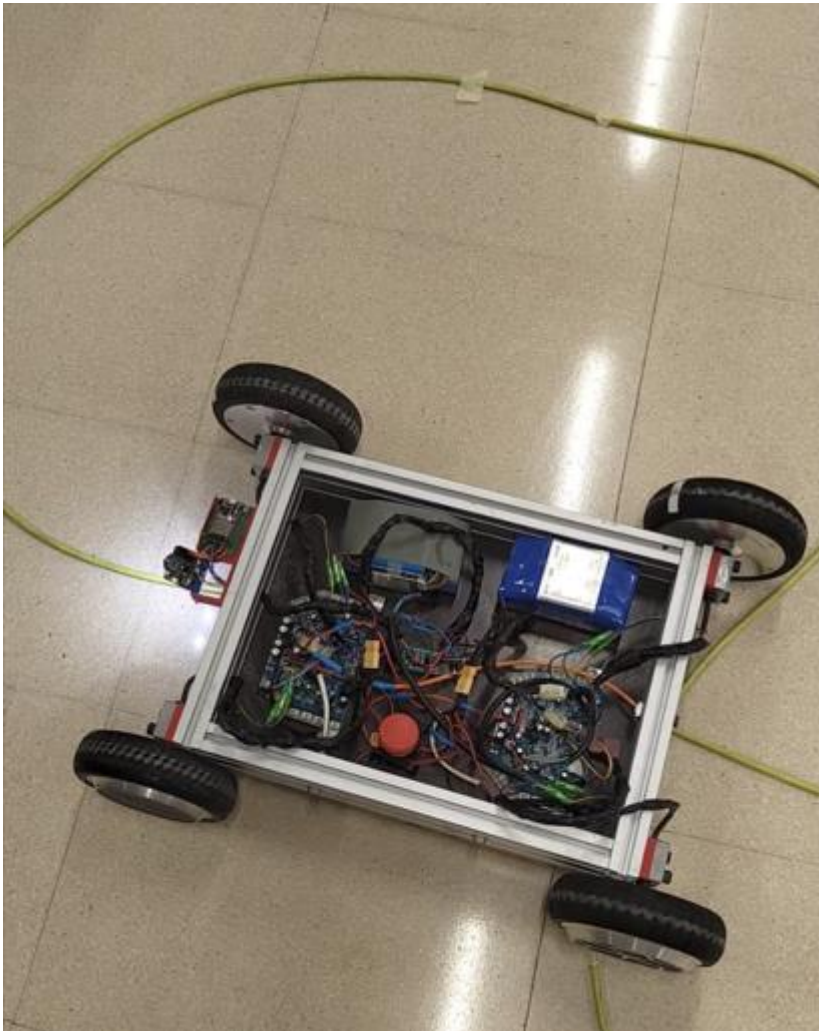


Figura 51. AGV sorteando intersección

Durante las pruebas se logró que el robot realizase curvas de 180° de hasta 30 cm de radio aproximadamente. En la figura 51 podemos ver al robot realizando una ruta circular. Además, mostró un buen comportamiento ante intersecciones de líneas perpendiculares, lo que permite que el trazado de la ruta tenga cruces.

Una vez comprobado el correcto funcionamiento del robot sin peso, se comenzaron las pruebas de carga para las que se utilizaron pesas de 12.25 Kg

(figura 52). De esta forma se puede evaluar el comportamiento del robot con diferentes pesos sin poner en peligro el *shaker* que transportará en un futuro.

El comportamiento del robot fue ideal hasta superar los 25 Kg de peso, entonces comenzó a presentar inestabilidad en el movimiento de los motores, provocando movimientos erráticos.



Figura 52. AGV en modo automático con 51 Kg

Dado que este problema ya lo había tenido antes, se localizó rápidamente su origen. El aumento de carga supone un mayor consumo de corriente y esto provoca unas mayores interferencias electromagnéticas. Una vez reforzado el apantallamiento de los cables se pudo continuar con las pruebas de carga sin problemas.

EL robot fue probado con diferentes pesos hasta un máximo de 75.5 Kg de lastre adicional, con los que se ha obtenido resultados satisfactorios (figura 53).



Figura 53. AGV en modo manual con 75.5 Kg

Sin embargo, estas pruebas revelaron algunas limitaciones del AGV:

- Dificultades en el modo automático:

Como es lógico, una carga mayor supone un aumento de las inercias en el robot. Además, dada la configuración con la que se han reprogramado las placas, ante una resistencia elevada los motores realizan movimientos ligeramente más bruscos.

Estos movimientos sumados a la inercia si bien no lo impedían, dificultaban la tarea del robot de seguir la línea de forma automática. Sin embargo, este sólo es un problema de ajuste del PID del robot, por lo que una vez esté listo para usarse con el *shaker* sólo habrá que realizar una calibración para mejorar el movimiento.

- Problemas de alimentación:

Durante la realización de estas pruebas se han detectado problemas debidos a picos de corriente durante el traslado de grandes pesos.

Por encima de los 50 Kg, tras un tiempo de uso del vehículo, éste se apagaba al realizar ciertos movimientos. Este problema es debido a que una de las baterías a duras penas es capaz de suministrar la corriente necesaria para mover la carga. Esto provoca que la placa que es alimentada por ella se apague y requiera de un reinicio.

La utilización de rampas de aceleración permitió disminuir el problema, pero no llegar a eliminarlo para cargas elevadas.

Dado que los *hoverboard* deben ser capaces de mover a personas adultas, las cuales suelen pesar más de 50 Kg, estas baterías deberían aguantar los picos de corriente provocados por los motores. Sabiendo esto se investigó el problema para esclarecer el asunto.

Finalmente se concluyó que era un problema exclusivo de una de las baterías la cual probablemente se haya deteriorado con el tiempo o que estuviese defectuosa de fábrica.

En el capítulo 8 se hablará de posibles sustitutos para las baterías, lo cual permitiría solucionar el problema.

- Rugosidad del suelo:

El diseño de cuatro ruedas y cuatro motores de este vehículo le permite realizar giros sobre sí mismo lo cual aporta una gran maniobrabilidad, sin embargo, implica que las ruedas deben deslizar sobre el suelo para poder girar.

Si bien esto no es un problema para el robot por sí mismo, cuando es lastrado aumenta enormemente la fuerza de rozamiento con el suelo, dificultando sus giros, aunque aumentando su tracción.

Esto, sumado al problema de alimentación mencionado anteriormente, supone que la generación de picos de corriente al intentar girar en suelo rugoso reinicie la placa.

Esto significa que la buena maniobrabilidad en el suelo deslizante de baldosas del laboratorio desaparezca en un suelo rugoso como puede ser el de la calle.

Durante las pruebas se ha comprobado que el límite de peso que puede cargar el robot para girar satisfactoriamente en suelo rugoso es de unos

25 Kg. Por encima de ese peso la batería defectuosa provocará una desconexión de dos de los motores.

El robot sin embargo sí parece capaz de girar con mayor peso en suelo rugoso. Con una alimentación más adecuada no debería haber problemas de maniobrabilidad en la mayoría de las superficies.

A través del siguiente enlace o del código QR se puede acceder a un archivo on todos los videos sobre la realización de las pruebas y el funcionamiento del AGV con diferentes cargas:



Figura 49. Miniatura del video



[Videos del AGV](#)

CAPITULO 8 CONCLUSIONES

8.1 Conclusiones

En este capítulo, se analizarán los resultados obtenidos, se comprobará el cumplimiento de los objetivos propuestos y se propondrán mejoras para continuar desarrollando las prestaciones del AGV.

8.1.2 Cumplimiento de objetivos

Los objetivos propuestos para este proyecto fueron los siguientes:

- Desarrollar un sistema de guiado robusto. Se ha logrado crear un sistema capaz de detectar y seguir la ruta establecida en diferentes condiciones de luminosidad y terreno. El vehículo finalmente ha podido seguir las rutas marcadas sin problemas y con independencia del entorno.
- Lograr un control satisfactorio de los motores de la máquina. Se ha conseguido un par motor adecuado a bajas revoluciones, permitiendo al AGV transportar de forma lenta y segura hasta 75.5 Kg durante las pruebas. No se ha seguido elevando la carga por falta de necesidad, pero es muy probable que el vehículo sea capaz de llevar pesos por encima de 100 Kg.
- Realizar una construcción segura y fiable de la máquina. Se han implantado las medidas de seguridad necesarias tales como seta de seguridad, interruptores para el apagado del robot y limitaciones software de la velocidad que impiden al AGV desplazarse por encima de 1.5 Km/h.
- Diseñar un modo manual remoto que permita maniobrabilidad. Se ha desarrollado satisfactoriamente un modo manual a través de un control remoto bluetooth ejecutado desde dispositivos inteligentes (móvil, tablet...). Este modo permite al operario maniobrar el vehículo, activar y desactivar el modo automático y controlar la velocidad.
- Implementar una configuración sencilla y versátil. Se ha logrado crear una configuración sencilla mediante la cual cualquier persona puede con facilidad manipular la cámara. De esta forma,

pueden cambiarse de forma sencilla los objetivos de seguimiento del robot, para cambiar de cable o calibrar en función de la luz y el terreno. El modo manual también es muy sencillo para que cualquiera pueda manejar el robot.

Por lo tanto, el proyecto ha logrado cumplir satisfactoriamente con todos los objetivos propuestos logrando crear un prototipo funcional de AGV para cargas ligeras y de precio económico.

8.1.3 Competencias adquiridas

Algunas de las competencias adquiridas durante el desarrollo del proyecto son:

- Aprendizaje sobre el control de motores Brushless mediante el método FOC y la electrónica implicada en ello.
- Descubrimiento de nuevas herramientas de programación tales como el programa *Virtual Studio Code* con la extensión *Platform.IO* que permiten programar un amplio abanico de dispositivos, así como un mayor control sobre las librerías y gestión de los proyectos.
- Conocimientos sobre los STM32, una familia de microcontroladores ampliamente utilizados en gran variedad de aplicaciones, incluyendo dispositivos embebidos, sistemas de control, dispositivos médicos, automóviles, y más. Así como el manejo de nuevos programas necesarios para su utilización como *STM32TrustedPackageCreator* y *STM32CubeProgrammer*.
- Profundización sobre los conocimientos adquiridos en la carrera en el ámbito de la visión artificial mediante la investigación de las cámaras candidatas a utilizarse en el proyecto y su posterior configuración e instalación.
- Perfeccionamiento de los conocimientos sobre diseño e impresión 3D necesarios para la fabricación de piezas clave para la construcción del proyecto, desde su fase de prototipado hasta el modelo final.
- Aprendizaje sobre protocolos de comunicación y control de forma práctica tales como I2C, UART, PWM, PPM, señales analógicas etc.

- Conocimientos prácticos sobre sistemas de lazo cerrado y la implementación de controladores PID.
- Aprendizaje sobre los experimentos realizados en el departamento de estructuras
- Conocimientos sobre los instrumentos y equipos utilizados para el desarrollo de los proyectos de investigación del laboratorio.

Uno de los mayores retos del proyecto ha sido la investigación y manipulación de las placas de *hoverboard*. La escasa información disponible y la gran variedad de modelos diferentes entre sí ha dificultado en gran medida el trabajo. Ha complicado especialmente el proyecto los códigos de colores utilizados para los cables de estas placas ya que no parecen seguir un patrón, ni siquiera para cables como los de alimentación.

Otros retos importantes han sido aquellos referentes a los problemas típicos de sistemas mecatrónicos (picos de corriente, alimentación insuficiente, interferencias, desajustes mecánicos...). Sin embargo, han logrado superarse para crear un prototipo funcional.

8.2 Líneas de mejora

En este capítulo se detallan las propuestas de mejora para optimizar y expandir los resultados alcanzados en este trabajo de fin de grado, identificando áreas clave para implementar cambios significativos.

8.2.1 Fuente de alimentación

Para este proyecto se ha escogido como fuente de alimentación las dos baterías de 36 V de *hoverboard* tal y como se explica en la selección de componentes.

Sin embargo, aunque es una alimentación válida para este vehículo tiene una serie de inconvenientes que podrían ser resueltos en el futuro.

El principal es la seguridad ya que este tipo de baterías son conocidas por no ser demasiado estables. Durante la realización del trabajo previo a este proyecto las baterías del robot fallaron provocando que el vehículo se incendiase y hubiese que comenzar de cero.

La utilización de baterías de mayor calidad u otro tipo de alimentación es un frente abierto con el que se podría mejorar el diseño del robot.

El segundo problema es el ya mencionado en el capítulo 7. La reducida capacidad de descarga de una de las baterías provoca fallos en la alimentación con cargas elevadas.

Bien por diseño, defecto de fábrica o deterioro esa batería no es apta para los propósitos de este proyecto y su sustitución es necesaria.

El último problema es que el *shaker* que debe trasladar el AGV también requiere de una alimentación y no sería válido alimentarlo a través de las baterías ya que requiere demasiada potencia.

La solución que se plantea sería la de instalar una fuente de alimentación en el robot que recibiese la corriente del mismo origen que la alimentación del *shaker*. Esta fuente permitiría alimentar de forma segura y eficiente ambas placas simultáneamente.

Esto supondría mantener al robot permanentemente conectado a través de un cable lo que reduciría sus prestaciones inalámbricas, sin embargo, dado que el *shaker* debe estar alimentado mediante un cable de todas formas, no sería un inconveniente.

8.2.2 Mando personalizado

Otra posible mejora del proyecto sería el diseño y fabricación de un mando personalizado para el AGV.

La utilización de una aplicación móvil gratuita es una solución válida y sencilla, pero con pocas prestaciones. Cuenta con limitaciones como sólo poder establecer 3 velocidades fijas ya que este mando sólo cuenta con 4 botones. Otra limitación es la imposibilidad de indicar al usuario si el robot se encuentra en modo manual o automático.

El último y principal problema del control mediante dispositivo móvil es el limitado rango de distancia de la señal emitida por el bluetooth.

Actualmente está pensado para que un operario inicie la marcha desde un extremo y otro retome el control desde el otro, sin embargo, por temas de seguridad sería conveniente mantener el gobierno de la máquina durante todo el trayecto.

Por ello se plantea como posible mejora el diseño de un mando de control remoto que incluya funciones como informar del estado del vehículo al operario, controlar la máquina mediante un “joystick”, regular la velocidad mediante un potenciómetro y que sea capaz de mantener la señal a lo largo de toda la ruta (mediante un módulo de radio, por ejemplo).

Otra posibilidad sería la de hacer un mando de control manual no remoto si se cambia la alimentación como se ha indicado anteriormente. Dado que probablemente sea necesario mantener conectados unos cables al vehículo de forma permanente se podría añadir uno más para su control.

8.2.3 Seta de emergencia remota

El principal argumento para la realización de un nuevo mando de control es el de mantener la seguridad a lo largo de todo el trayecto. Pudiendo detener el robot en cualquier momento.

Para poder hacer esto se plantea la posibilidad de incorporar una seta de emergencia remota.

Esto se realizaría de forma que la seta estuviese en manos de los operarios mientras el robot realiza su ruta. En caso de emergencia y de forma inalámbrica, al pulsar la seta enviaría una señal al robot a través de un módulo de radio, para cortar la alimentación y detenerlo.

Por otra parte, en el vehículo iría instalado un relé a través del cual pasa la alimentación del AGV. Ese relé estaría conectado al módulo de radio receptor y ambos a una fuente de alimentación independiente de la del vehículo.

De esta manera, garantizamos que si sucede algún fallo en las baterías del robot el módulo de radio y el relé siguen funcionando para poder cortar la alimentación.

Otra posibilidad es la de hacerlo de forma alámbrica si finalmente el robot se mantiene conectado por cables en todo momento.

8.3 Consideraciones adicionales

8.3.1 Estimación de horas

A continuación, se muestra la tabla 1 con la distribución aproximada de horas dedicadas a cada una de las fases del proyecto.

FASES DEL PROYECTO	HORAS
Selección de los componentes	15
Construcción del prototipo	15
Investigación sobre los hoverboard y reprogramación	45
Selección del protocolo de comunicación con las placas	90
Construcción del circuito	35
Programación	25
Solución de problemas	50
Implementación y pruebas	20
Ajustes y perfeccionamiento	15
Redacción de la memoria	60
TOTAL	370

Tabla 1

CONCLUSIONES

8.3.2 Estimación de costes

A continuación, se realizará una estimación de los costes totales del proyecto, para el cual se elaborará una lista con los componentes utilizados y sus respectivos costes en las tablas 2 y 3.

Costes del prototipo:

MATERIALES	CANTIDAD	PRECIO (€)	COSTE (€)
Material de impresión (PLA)	106 g	15€/Kg	1.59
Motor 5 V	4.00	0.90 €	3.60
Controlador L298N	1.00	4.00 €	4.00
Porta pilas	1.00	2.00 €	2.00
Pilas AA	4.00	0.75 €	3.00
Interruptor	1.00	1.30 €	1.30
ESP32	1.00	5.00 €	5.00
Pixy cam 2.1	1.00	75.30 €	75.30
Cableado y tornillería			0.30
		TOTAL	96.09 €

Tabla 2

IMPLEMENTACIÓN DE TÉCNICAS ROBUSTAS DE GUIADO DE VEHÍCULOS AUTÓNOMOS BASADAS EN VISIÓN ARTIFICIAL

Costes del AGV:

MATERIALES	CANTIDAD	PRECIO (€)	COSTE (€)
Perfiles Fasten	4.00	15 €/Kg	16.00
ESP32	1.00	5.00	5.00
Tablas	2.00	4.00	14.00
<i>hoverboard</i>	2.00	40.00	80.00
Material de impresión (PLA)	91 g	15 €/Kg	3.00
Interruptor dos canales	1.00	1.50	1.50
ESP32	1.00	5.00	5.00
Seta de emergencia	1.00	17.85	17.85
Level shifter	1.00	0.80	0.80
Regulador de tensión L7805	2.00	0.69	0.69
Placas perforadas	2.00	0.43	0.86
Cinta aislante (10 m)	2.00	1.50	3.00
Cinta adhesiva de aluminio	1.00	4.00	4.00
Pixy cam 2.1	1.00	75.30	75.30
Tornillería, estaño, termo retráctil y cableado			4,00
		TOTAL	231.00 €

Tabla 3

Cabe a destacar que parte de los componentes utilizados en el prototipo luego han sido reutilizados en el modelo final. Un ejemplo de esto es la cámara, siendo una de las piezas más costosas.

Por lo tanto, el coste total del proyecto sería de unos 246.49 € aproximadamente.

8.3.3 Impactos positivos del proyecto

El precio del AGV es muy económico en comparación con productos similares del mercado (los cuales superan los miles de euros). Si bien se trata de un robot pequeño y con poca capacidad de carga comparado con sus semejantes, puede ser de gran utilidad en determinadas situaciones. Su configuración sencilla, tamaño y peso reducido y su precio económico lo hacen ideal para situaciones en las que se tenga que mover una carga de forma regular. La gran versatilidad de este vehículo y sus componentes hacen que sea fácilmente adaptable a todo tipo de situaciones.

La gran mayoría de sus piezas son recicladas, contribuyendo al cuidado medioambiental. Especialmente hay que atender a la reutilización y correcto reciclado de materiales como las baterías, cuya mala gestión puede provocar grandes daños medioambientales.

La utilización de este equipo para la carga y desplazamiento de grandes pesos puede contribuir de forma positiva en la salud de las personas. Aunque el rango de pesos que este AGV es capaz de trasladar es cercano al que podría mover una persona con la herramienta adecuada (una carretilla, por ejemplo), cargar continuamente con pesos de esa magnitud puede causar lesiones en las personas. La sencillez de este vehículo y su bajo coste lo hacen rentable para liberar a las personas de cargar con pesos elevados en situaciones en las que sea algo habitual.

BIBLIOGRAFÍA

Nota: todas las referencias a internet han sido accedidas por última vez en junio de 2024.

Referencias y fuentes:

[1]: F. Fonseca Porras, *Vehículo multifuncional de apoyo a pruebas de carga en ingeniería*, Trabajo Fin de Grado, Universidad de Valladolid, 2023.

[2]: <https://www.edsrobotics.com/agv-robot/>

[3]: <https://www.zonaoutdoor.es/whinck-hoverboard-lg-rojo>

[4]: <https://github.com/EFeru/hoverboard-firmware-hack-FOC>

[5]: <https://github.com/EFeru/hoverboard-firmware-hack-FOC>

[6]: <https://www.tiendatec.es/raspberry-pi/gama-raspberry-pi/2149-raspberry-pi-5-modelo-b.html>

[7]: <https://www.amazon.es/XVTX-ESP32-CAM-Bluetooth-ESP32-CAM-MB-compatible/dp/B093GSCBWJ>

[8]: <https://openmv.io/>

[9]: <https://tienda.bricogeek.com/sensores-imagen/1410-huskylens-camara-de-vision-artificial.html>

[10]: <https://pixycam.com/>

[11]: <https://pixycam.com/>

[12]: https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:port_pinouts

[13]: <https://www.az-delivery.de/es/products/esp32-developmentboard>

[14]: <https://www.az-delivery.de/es/products/esp32-developmentboard>

[15]: <https://www.amazon.es/Bater%C3%ADa-lones-Litio-Hoverboard-Reemplazo/dp/B0C4H17DQB>

[16]: <https://es.rs-online.com/web/p/botones-pulsadores-de-parada-de-emergencia/1682546>

[17]: <https://www.amazon.es/spannungsregler-220-L7805-7805-positiva-Regulador-regulators/dp/B07H732W8B>

[18]: <https://my.cytron.io/c-jetson-camera/c-raspberry-pi-cm/c-raspberry-pi-c-raspberry-pi-hat-accessories/p-4-channel-logic-level-shifter-pre-soldered-headers>

[19]: <https://www.amazon.es/PATIKIL-Circuitos-Prototipo-Perforada-Electr%C3%B3nico/dp/BOCBPTG39M>

[20]: <https://pixycam.com/>

[21]: <https://github.com/lucysrausch/hoverboard-firmware-hack>

[22]: <https://www.az-delivery.de/es/products/st-link-v2-mini-simulator-download-programmer-stm8-6-stm32>

[23] Cifre Gregorio, Sergio, *Control FOC de motores BLDC*, Trabajo Fin de Grado, Universidad de Valencia, 2020.

[24]:

<https://www.didacticaselectronicas.com/index.php/semiconductores/convertidores-1/frecuencia-a-voltaje/pwm-voltaje-0-10v-detail>

[25]: <https://cricardoromani.wordpress.com/arduino/1o-year/explicaciones-previas/explicacion-de-las-entradas-salidas-pwm/>

[26] Solectroshop. s.f. <https://solectroshop.com/es/blog/que-es-pwm-y-como-usarlo--n38>

[27] laboratorioluon. s.f. <https://www.laboratorioluon.com/generar-senal-pwm-para-servo-con-avr-atmega328p/>

[28]: <https://www.laboratorioluon.com/receptor-rc-con-arduino-moviendo-servos-con-pwm/>

[29] Johnson, Michael A., dir.; Moradi, Mohammad H, *PID control: New Identification and Design Methods*, Edición 2005, London: Springer.

[30]: <https://www.picuinio.com/es/control-pid.html>

ANEXOS

Anexo A: Diseños 3D.

Anexo B: Códigos desarrollados.

Anexo C: Esquemas eléctricos.

Anexo D: Lista de componentes y estimación de precios.