ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

CREACIÓN DE UN ENTORNO INTERACTIVO MEDIANTE ROBOTS PARA LA CONCIENCIACIÓN DE AYUDA A LA DISCAPACIDAD

Autor: Rocío Baixas Durbán
Director: José San Martín López

Madrid
Julio 2020

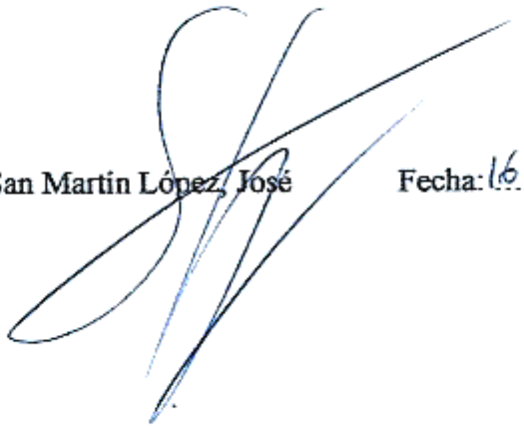
Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título "Creación de un entorno interactivo mediante robots para la concienciación de ayuda a la discapacidad" en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2019/20 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Baixas Durbán, Rocío Fecha: 15./07./2020

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: San Martín López, José Fecha: 16./07./2020

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

CREACIÓN DE UN ENTORNO INTERACTIVO MEDIANTE ROBOTS PARA LA CONCIENCIACIÓN DE AYUDA A LA DISCAPACIDAD

Autor: Rocío Baixas Durbán
Director: José San Martín López

Madrid
Julio 2020

CREACIÓN DE UN ENTORNO INTERACTIVO MEDIANTE ROBOTS PARA LA CONCIENCIACIÓN DE AYUDA A LA DISCAPACIDAD

Autor: Baixas Durbán, Rocío.

Director: San Martín López, José.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

En el proyecto, se pretenderá el diseño de un robot programable con el objetivo de que este pueda simular ciertas discapacidades. El fin del proyecto es que, con el tiempo, este tipo de robots se puedan introducir en las dinámicas grupales cuyo objetivo es luchar contra la discriminación dirigida a las personas que tienen alguna discapacidad sensorial o física.

INTRODUCCIÓN

Actualmente, sigue existiendo discriminación a personas que tienen algún tipo de discapacidad, ya sea social o profesional. En el mundo moderno que está constantemente avanzando, también tienen que evolucionar los métodos de concienciación para que sigan dando resultado. Los robots son una herramienta nueva y recreativa que también pueden ser introducidos en las dinámicas de grupo para la concienciación y sensibilización de las personas contra la discriminación.

El proyecto se ha realizado en base a la premisa "discapacidad no significa incapacidad". En este proyecto se pretende diseñar un robot que funcione como avatar para el usuario o participante de la concienciación, y que pueda poner a prueba las actividades más cotidianas mientras que el robot simula una discapacidad. El proyecto tiene como fin que las actividades se realicen siempre, pero poniéndose en la posición de las personas que tienen esa discapacidad. De esta manera, se piensa que los participantes serán capaces de empatizar con mayor facilidad.

OBJETIVOS

El objetivo principal del proyecto es el de realizar un robot que sea capaz de simular ciertas discapacidades. Las discapacidades que se han conseguido simular a partir del robot realizado son la pérdida total de la vista, el daltonismo y una discapacidad motora en piernas y en brazos. Este objetivo se divide en las siguientes metas:

- Diseño de una plataforma robotizada móvil que permita al usuario moverse por el entorno a modo de avatar.
- Diseño de un brazo robotizado manejable por el usuario con control remoto.
- Programación del robot en las diferentes simulaciones.
- Elección de los componentes electrónicos.
- Elaboración de la memoria que sirva de documentación del proceso.

Otro objetivo secundario que tiene este proyecto es el de intentar sustituir los materiales que tienen mayor impacto en el medio ambiente por unos objetivos más ecológicos y menos dañinos.

También se pretende conseguir alinear el proyecto con al menos uno de los 17 Objetivos de Desarrollo Sostenible (ODS) de Naciones Unidas. Cabe destacar, que en caso de que se consiguiesen ambos propósitos mencionados, el proyecto estaría alineado con dos de estos objetivos.

DESARROLLO

En primer lugar, se hizo una elección sobre los componentes necesarios para realizar el proyecto. Entre esos elementos se eligió el robot Edison V2.0 como robot plataforma para así facilitar el movimiento del robot en el plano horizontal. También, se decidió elegir el módulo ESP32-CAM como unidad de control. Este posee comunicaciones Wifi y bluetooth. Los demás elementos electrónicos que se consideraron necesarios para el proyecto son un servomotor, un motor paso a paso, un controlador para limitar la corriente por el motor paso a paso y baterías recargables para su alimentación.

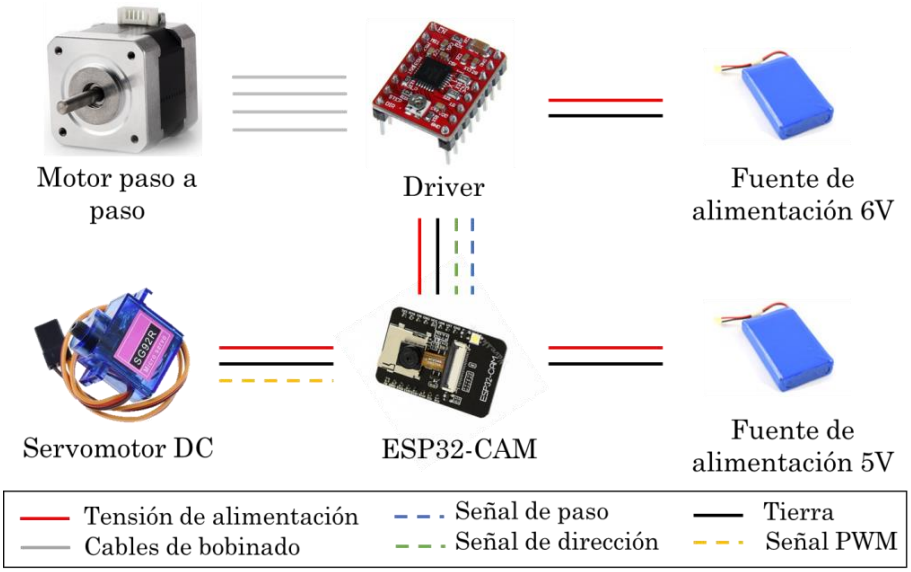


Ilustración 1. Conexión de elementos electrónicos.

Se decidió el diseño del brazo robot. Se concluyó que el brazo constaría de dos grados de libertad en total, permitiendo mover el brazo en el eje Y, y abrir y cerrar los dedos en de la pinza. Las partes están diseñadas para no soportar mucho esfuerzo, y así poder ser elaboradas a partir de aluminio o de múltiples capas de cartón gris.

También se realizó la programación completa de la tarjeta ESP32-CAM. Se programó un servidor web a partir con el cual, el usuario será capaz de dirigir el brazo del robot desde cualquier dispositivo que tenga internet.

En la programación del servidor web existe la posibilidad de controlar tanto el movimiento del brazo robot como la imagen enviada desde la cámara, cambiando el color de la imagen o parando la visualización de imagen por completo.

RESULTADOS

El diseño de la pinza consta de dos dedos que podrán abrir la pinza una amplitud de 5 centímetros y levantar un peso de hasta 250 gramos. Se puedo realizar una simulación cinemática y se comprobó un correcto funcionamiento.

El resto del brazo se compone finalmente de 5 segmentos de diferentes medidas que también podrán levantar los 250 gramos de carga más el peso de la pinza y el

servomotor. Su funcionamiento cinemático también ha sido comprobado por medio de una simulación.

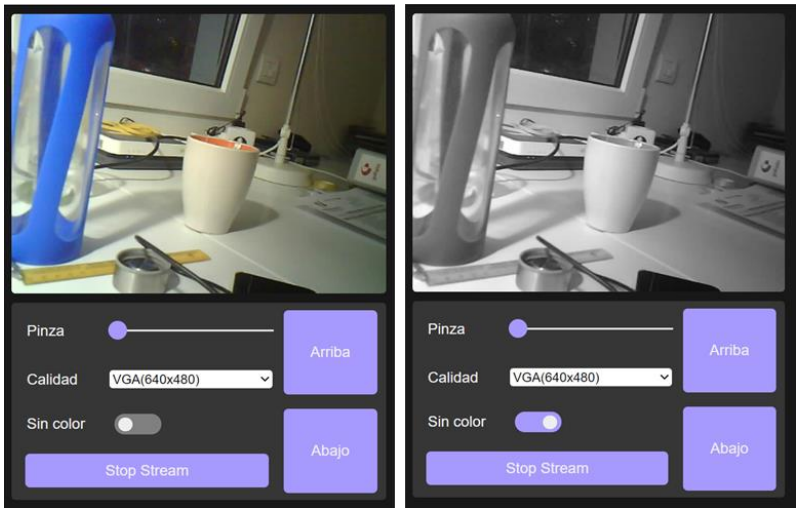


Ilustración 2. Diseño de servidor web.

La programación del conjunto ESP32-CAM pudo ser comprobada y los resultados fueron óptimos. En la Ilustración 2 se puede apreciar el diseño final del servidor web. Ofrece una pestaña que permite mejorar la calidad de la imagen, un interruptor que altera el color de la imagen y un botón que permite detener la transmisión de imagen. Para el control del brazo, ofrece un pasador para el control de la apertura de la pinza y dos pulsadores que subirán o bajarán el brazo mientras que me mantengan pulsados.

CONCLUSIONES

Se han podido llevar a cabo las simulaciones de pérdida de la visión, daltonismo y discapacidad motora de extremidades. Con esto se confirma el objetivo principal del proyecto que consistía en poder simular ciertas discapacidades para, posteriormente, servir en futuras dinámicas grupales de concienciación.

En segundo lugar, aunque no se haya podido hacer un proyecto enteramente ecológico, existen partes del proyecto diseñadas para ser construidas a partir de material reciclable o biodegradable.

En general se puede decir que se han conseguido alcanzar los objetivos propuestos. Las simulaciones se han conseguido desarrollar de manera correcta y las decisiones sobre el material se han tomado teniendo en cuenta el respeto al medio ambiente.

CREATION OF AN INTERACTIVE ENVIRONMENT WITH ROBOTS FOR CONSCIOUSNESS-RAISING ABOUT AID TO IMPAIRMENT

Autor: Baixas Durbán, Rocío.

Director: San Martín López, José.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

PROJECT SUMMARY

This project pretends the design of a robot able to simulate certain disabilities or impairments. The aim of this project is that with the passage of time, this kind of technology could be introduced in group dynamics which confront discriminations against people who suffer this kind of condition.

INTRODUCTION

These days, discrimination against people who suffer any type of disability or impairment still exists, including social and professional discrimination. In an environment where everything is evolving rapidly, consciousness-raising methods should not fall behind, and introduce new developments to guarantee results in the modern society. Robots are a new tool that could be introduced in group dynamics for the awareness about discrimination against disability.

The project is based on the premise that "disability does not mean incapability". This project pretends the design of a robot which will work as an avatar for the participant in the group dynamic and that can be remotely controlled. The user will need to achieve certain activities that will be complicated as the robot will be simulating a disability. This way, members are thought to feel empathy more easily.

OBJECTIVES

The main goal of this project is the realization of a design able to simulate certain disabilities. These disabilities are blindness, colorblindness, and a motor disability. This goal can be divided into smaller objectives:

- Design of a robotic platform controlled remotely that allows user to move in the environment as an avatar.
- Design of a robot arm remotely controlled.
- Programming of the robot in different situations.
- Election of the electronic components.
- The writing of a report that documenting the process.

A secondary goal for this project was the substitution of the materials that have a negative impact on the environment with other green materials.

In addition, it is intended to align the project with at least one of the 17 Sustainable Development Goals (SDGs) designed by the United Nations General Assembly. If the first two goals mentioned are achieved, then the project will accomplish two of the objectives proposed by the United Nations.

DEVELOPMENT

First, the electronic components needed for the project were chosen. Edison robot was chosen as the platform of the whole project, as it eases the movement of the robot in the horizontal plane. In addition, ESP32-CAM was selected as the control unit. The ESP32 includes a Wi-Fi and Bluetooth module, which will be essential for the remote control. Moreover, the final design also includes a servomotor, batteries, a stepper motor, and its driver.

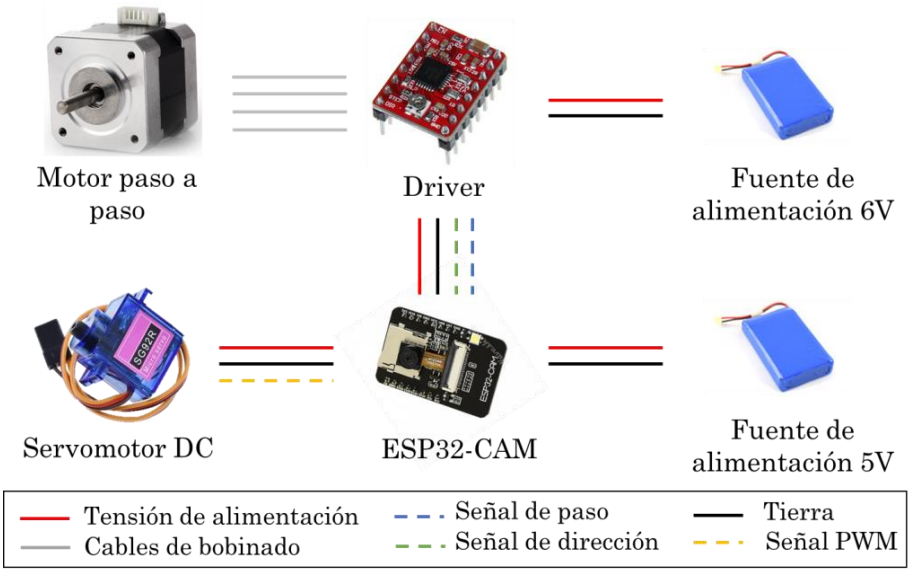


Figure 1. Electronic components attached to the ESP32-CAM.

The design of the robot arm was decided. Finally, the arm would consist of two degrees of freedom, allowing the movement on the vertical axis and the opening and closing of the claw. The whole structure is designed not to tolerate heavy weight; thus, the arm could be manufactured based on recycled and biodegradable materials, as aluminum or multiple layers of grey chipboard.

The complete programming for the ESP32-CAM was written. It was planned to program a web server, allowing anyone with an internet-connected device to stablish a connection with the robot.

The program it is thought to control the arm's movement and the image recorded by the camera, variating the color, or stopping the image display if requested.

RESULTS

The design of the claw can open 5 centimeters and raise a weight of 250 grams. It was possible to simulate the movement of the claw with the Solid Edge program.

The rest of the arm is finally composed of 5 segments of different lengths that will also be able to raise the 250 grams previously planned, plus the weight of the claw and

the servomotor. The movement of the whole structure was simulated, and the movement resulted correct.

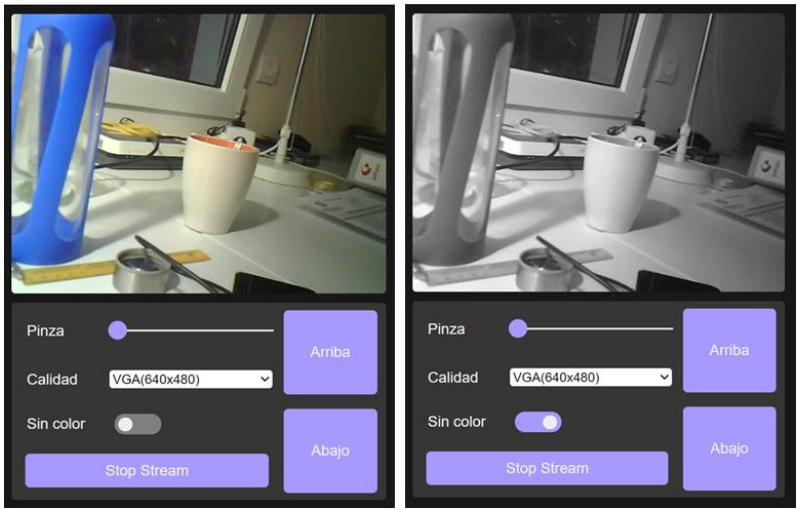


Figure 2. Web server design.

The ESP32 code was proven and the results were correct. Figure 2 shows the final design of the web server. It offers different elements to improve the resolution, a switch which can modify the colors of the image and a button which stops the display of the image. For the arm control, there is a fastener por que opening of the claw and two push buttons that will raise and lower the arm when active.

CONCLUSIONS

It was possible to accomplish the simulation of blindness, colorblindness, and motor disability, which confirms the realization of the principal objective, that consisted of the simulation of different disabilities for the following application in dynamic groups.

Secondly, although it is not been possible to elaborate a completely ecologic robot, the elements of this project were designed for its construction with recycled or biodegradable materials.

In conclusion, the aims are considered to be achieved as the simulations have been developed correctly and the decisions made on materials were based on the well-being of the environment.

Índice general

Índice de figuras

Índice de tablas

Siglas

GPIO	*General-Purpose Input/Output*
HTTP	*Hypertext Transfer Protocol* (Protocolo de transferencia de hipertexto)
IR	Infrarrojos
I2C	*Inter-intergrated Circuit* (Circuito Inter-Integrado)
I2S	*Integrated Interchip Sound*
RC	Radio Control
RX	Recepción
STEAM	*Science, Technology, Engineering and Mathematics* (Ciencia, Tecnología, Ingeniería y Matemáticas
PCM	*Pulse Code Modulation* (Modulación por el código de pulso)
PPM	*Pulse Position Modulation* (Modulación por posición del pulso)
PWM	*Pulse Width Modulation* (Modulación por anchura de pulso)
TCP	*Transmission Control Protocol* (Protocolo de control de transmisión)
TTL	*Transistor Transistor Logic* (Lógica transistor a transistor)
TX	Transmisión
UART	Universal Asynchronous Receiver/Transmitter
URI	*Uniform Resource Identifier* (Identificador de recursos uniforme)

1. Introducción

En 2008, el Instituto Nacional de Estadística[1] llevó a cabo su último estudio dirigido a las personas que tienen alguna discapacidad, concluyendo que un 8,55% de la población española, es decir, 3,85 millones de personas, posee algún tipo de discapacidad.

La encuesta dedica un apartado para hablar de la discriminación que sufren estas personas resaltando que un 1,2 % de este grupo ha sufrido este tipo de comportamiento abusivo debido a sus limitaciones. Asimismo, se considera que sólo un 7,7% de las personas discriminadas lo han denunciado. Según la encuesta "*ésta se produce principalmente en las relaciones sociales, en el entorno de participación social y en la atención sanitaria, por este orden*". En la siguiente figura se representa la discriminación sufrida dependiendo del tipo de discapacidad.

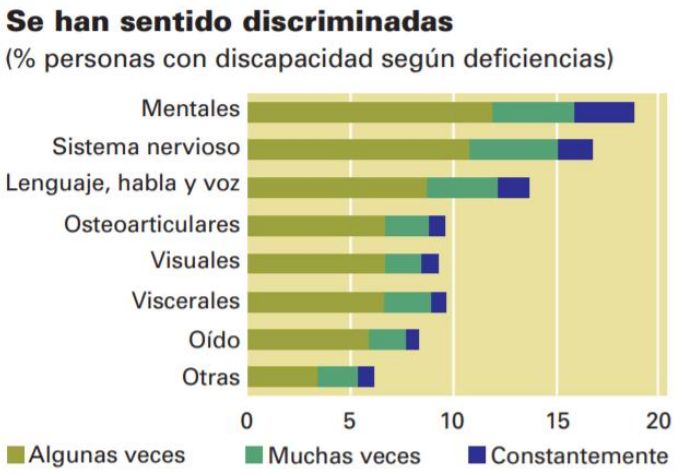


Figura 1. Personas discriminadas según deficiencias.

Puesto que este problema sigue afectando a gran cantidad de personas en el país y globalmente, es necesario modernizar las soluciones que se han dado hasta ahora. Por ello, este proyecto introducirá el uso de robots en las dinámicas de concienciación.

1.1. Motivaciones

Para luchar contra la discriminación social la medida más efectiva es la sensibilización. Es necesario concienciar a la sociedad, tanto niños como adultos, para que se comprendan las necesidades y dificultades que afrontan a diario estas personas. Además, entre adultos es necesario introducir la idea que "*discapacidad no significa incapacidad*" [2] para fomentar la inserción laboral de este grupo.

Una de las técnicas más empleadas para la sensibilización es la dinámica de grupo. En ellas se plantean ciertos interrogantes y se promueve que, como grupo, se busque una solución a estos. El fin de la dinámica grupal siempre es el aprendizaje por medio de la interacción social por lo que es imprescindible la participación de los integrantes.

Por otro lado, en los últimos años se ha fomentado una nueva forma de educación, llamada STEAM[3]. Busca motivar la curiosidad e imaginación de sus participantes y promueve que se busquen varias soluciones para un mismo problema. Este modelo complementa cinco disciplinas (ciencia, tecnología, ingeniería, arte y matemáticas), que se aplican de manera experimental y no teórica. Para la implementación de esta metodología en un proyecto se deben definir los objetivos y dejar que el propio concursante busque sus propias soluciones a partir de los medios de los que dispone.

Esta nueva disciplina ha mostrado tener múltiples ventajas frente a la enseñanza tradicional, ya que, para empezar, los estudiantes presentan una actitud activa y retienen mejor la información aprendida. Otra de las ventajas que se pretende explotar en este proyecto es la experimentación en primera persona. Al tratarse de una concienciación, es necesario que los participantes empaticen con las personas discapacitadas, por tanto, es necesario un contacto que cause un impacto lo más profundo posible. Es esta la razón por la que se piensa que este modelo debería ser imitado en este tipo de dinámicas para atraer la atención de los implicados.

En resumen, este proyecto pretende introducir las nuevas tecnologías en las dinámicas de grupo para así atraer y fomentar la participación de la gente buscando un ambiente relajado y recreativo sin alterar el objetivo de la dinámica: la lucha contra la discriminación de personas con alguna discapacidad.

1.2. Objetivos

El objetivo principal de este proyecto es el diseño de un robot que tenga incluidas las inteligencias necesarias para simular ciertas discapacidades y que cumpla su fin de ayudar a la concienciación de estas. Con el fin de completar este objetivo, el proyecto se deberá dividir en metas más pequeñas. Estas son las siguientes:

- Diseño de una plataforma robotizada móvil que permita al usuario moverse por el entorno a modo de avatar.
- Diseño de un brazo robotizado manejable por el usuario con control remoto.
- Programación del robot en las diferentes simulaciones.
- Elección de los componentes electrónicos.
- Elaboración de la memoria que sirva de documentación del proceso.
- Elección de materiales basados en la baja incidencia ambiental como ayuda al medio ambiente. Esto se desarrollará en el siguiente apartado.
- Alineación del proyecto con los Objetivos de Desarrollo Sostenible de Naciones Unidas. Esto quedará explicado en el Anexo A del documento.

1.2.1. Objetivo ecológico

Muchos componentes para robots con fin recreativo están compuestos por plástico. Aunque muchos tipos de plástico son reciclables, la gran mayoría resulta incinerado o desechado. Esto se debe a que el reciclaje del plástico no presenta beneficio económico.

En primer lugar, la gran variedad existente de plásticos dificulta el proceso de reciclaje de este material, ya que cada tipo requiere un proceso de reciclaje diferente haciendo necesaria la clasificación de los residuos antes del proceso. Además, a diferencia de otros materiales como el aluminio, el plástico es un material que presenta pérdidas de calidad durante el trascurso de su vida útil, por tanto, el reciclado de este material siempre resultará en componentes de peor calidad en comparación al plástico virgen.

Estas son sólo algunas de las razones por las que se decide no reciclar el plástico y terminar su ciclo de vida después de su primer uso. Gracias al estudio realizado en 2019

por la asociación empresarial de Plastics Europe[4] se puede constatar que los tres posibles finales para todo plástico se reducen a incineración, vertedero o reciclaje y en qué porcentaje se elige cada una.

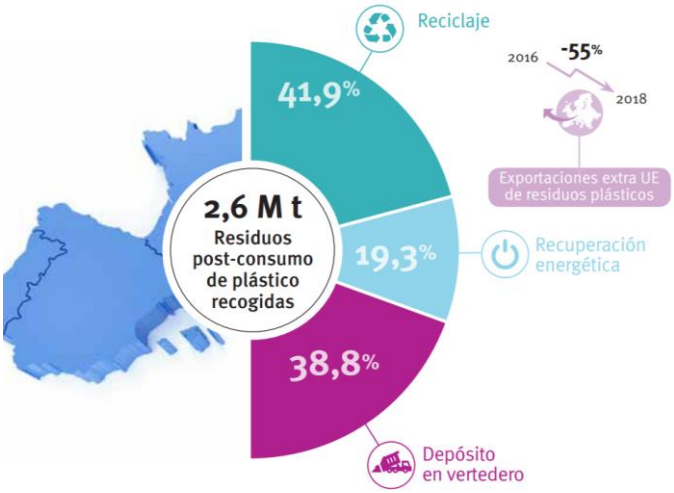


Figura 2. Tratamiento de residuos plásticos en España en 2018.

En la imagen anterior se presentan los datos sobre el destino de estos plásticos en España en el año 2018. Se puede apreciar cómo un 38,8% de los 2,6 millones de toneladas de residuo recogido es tirado en vertederos, siendo el tiempo de descomposición de una bolsa de plástico superior a 150 años. También cabe destacar, que la demanda de plástico de un sólo uso en España en 2018 llegaba a los 4,7 millones de toneladas[5], con lo que se puede concluir que el porcentaje de plástico recogido es muy pequeño comparado con el plástico que no se recoge.

Los problemas medioambientales que ocasiona el plástico invitan a dejar de depender de este material y buscar otras alternativas menos dañinas.

1.3. Funcionamiento del robot

En este caso, se creará un robot con diferentes modalidades que pueda simular ciertas discapacidades, como por ejemplo podrían ser, daltonismo o ceguera, sordera, autismo y deformidad en extremidades superiores e inferiores. Este robot funcionará como un avatar y será sometido a pruebas en las que se vea comprometido por alguna discapacidad. El objetivo del participante será buscar la manera de finalizar la prueba con éxito, demostrando así, que discapacidad no implica incapacidad, aparte de sentir las limitaciones que tiene alguien con estas discapacidades al realizar cierto tipo de tareas.

- **Simulación de ceguera.** En esta dinámica el robot perderá el sentido de la visión. El robot será colocado en un laberinto y el usuario, tendrá que buscar la salida sin poder utilizar el sentido de la vista. Para el desarrollo de la dinámica, el robot tendrá instalados unos sensores ultrasonidos y un altavoz para poder mandar una señal auditiva cada vez que se encuentre con un obstáculo.
- **Simulación de daltonismo.** En este segundo caso la cámara sí que estará activa, aunque la imagen tendrá el color distorsionado. De esta manera el usuario no podrá distinguir los colores que aparecerán en la pantalla con facilidad y tendrá que buscar la manera de poder seguir con el juego sin que le resulte un impedimento.
- **Simulación de disminución de las capacidades motoras inferiores.** En principio, la idea es que la velocidad del motor se vea disminuida. Según se vaya avanzando en el proyecto se intentará que dicho comportamiento del robot simule de manera más realista este tipo de limitación.
- **Simulación de disminución de las capacidades motoras superiores.** Se incluirá una dinámica en el que sea necesario el uso de los brazos para poder transportar objetos. En el robot, esta funcionalidad estará reducida, y el objetivo del usuario será conseguir el mismo fin pero por otros medios.

1.4. Recursos

En este apartado se explicarán los recursos que han sido necesarios para la realización del proyecto. Se incluye un apartado específico para los elementos Hardware (Sección 1.4.1.) y otro para los elementos Software (Sección 1.4.2.).

1.4.1. Recursos hardware

El proyecto se realizó a partir de un único ordenador portátil, Lenovo Yoga 720.

Figura 3. Lenovo Yoga 720.

Otros recursos hardware empleados son los siguientes:

- **Edison STEAM Robot V2.0.** Robot que se empleará como plataforma inicial para un movimiento por control remoto sobre un plano horizontal.
- **ESP32-CAM.** Placa con comunicaciones Wifi y Bluetooth. Se utilizará como unidad de control.
- **Cámara OV2640.** Cámara de 2 megapíxeles conectada a la ESP32-CAM.
- **Micro Servo Digital SG92R.** Su función es abrir y cerrar la pinza del brazo robot.
- **Motor paso a paso 42BYGH3 de *Kysan Electronics*.** Motor que moverá el brazo de arriba a abajo.
- **Controlador de motor paso a paso o Driver DRV8834.** Permite controlar la intensidad de corriente que circulará por el bobinado del motor paso a paso.
- **Baterías recargables de 5V y 6V** para la alimentación de la placa y del controlador del motor, respectivamente.

1.4.2. Recursos Software

Los recursos software empleados para este proyecto son los siguientes:

- **Entorno de programación de Arduino (IDE).** Para escribir el código de la unidad de control.
- **EdPy.** Aplicación online para programar el robot Edison V2.0 basada en lenguaje Python.
- **Solid Edge.** Para comprobar el funcionamiento de las estructuras mecánicas.
- **Autocad.** Para señalar las cotas de las simulaciones de Solid Edge.
- **iCiberChef.** Web que permite traducir códigos a otros lenguajes.

1.5. Contenido de la memoria

La memoria está dividida en diferentes secciones que describen el transcurso del proyecto y explica los resultados obtenidos en las pruebas finales. A continuación, se explica brevemente el contenido de cada sección.

El Capítulo 2 denominado "Estado del Arte" recoge brevemente los conocimientos necesarios para la elaboración de este diseño. Se recoge información sobre los diferentes tipos de control remoto y las diferentes comunicaciones en servidores web. Además, se exponen ejemplos de robots existentes que presentan cualidades parecidas a las que se pretenden con este proyecto.

En el Capítulo 3 o "Arquitectura del proyecto", se exponen los componentes que finalmente se emplearon en el proyecto y las comunicaciones que deben mantener entre ellos.

En el Capítulo 4 se elabora un estudio sobre materiales que se pueden emplear como alternativa al plástico.

En el Capítulo 5 se recoge el diseño mecánico del brazo robot. Además, se especifica la elección de componentes para su elaboración justificando las decisiones tomadas.

En el Capítulo 6 se explica la programación que se ha llevado a cabo para este proyecto. Se recoge una explicación, pero el código en sí se puede encontrar en el Anexo B.

En el Capítulo 7 se hace un estudio del consumo eléctrico y dependiendo de los resultados se eligen las fuentes de alimentación para el conjunto.

En el Capítulo 8 se recogen las pruebas que se han podido realizar para la comprobación del proyecto.

En el apartado denominado como "Apéndices" se recogen tres anexos. El Anexo A contiene la alineación del proyecto con los objetivos ODS. Los Anexos B y C contienen el código empleado para las simulaciones.

2. Estado del Arte

En el siguiente capítulo se describirán las técnicas de la actualidad que pueden ser útiles para el desarrollo del robot descrito. Entre ellas, se describen las características de los controles remotos de hoy en día, el envío de imagen en vivo y la posible alteración de dicha imagen. Además, se exponen robots de la actualidad que pueden servir como ejemplo para la realización del proyecto.

2.1. Control remoto

Los robots teledirigidos han tenido una gran cantidad de aplicaciones en los últimos años. Algunos ejemplos de estas numerosas aplicaciones son los vehículos teledirigidos o los drones. Sin duda, la aplicación más similar a lo que se pretende conseguir con este proyecto es la primera.

2.1.1. Radio Control

Actualmente, la mayoría de los vehículos teledirigidos con función recreativa funcionan con un tipo de comunicación RC (radio control)[6]. Este tipo de control se basa en una transferencia de datos unidireccional de emisor a receptor. Para ello, son necesarios dos tipos de protocolos: protocolo TX (comunicación entre TX y RX) y protocolo RX (comunicación entre RX y controlador). Entre los tipos de protocolo RX, destacan por su uso frecuente en los vehículos controlados la Modulación por Anchura de Pulso (PWM), la Modulación por la Posición de Pulso (PPM) y la Modulación por el Código de Pulso (PCM). En este proyecto se empleará únicamente la primera.

En la modulación de la anchura de pulso o *Pulse Width Modulation* (PWM)[7] se analiza el ciclo de trabajo de una señal cuadrada. Este ciclo de trabajo varía, modificando a su vez la tensión media de salida. Esta salida es modulada por el microcontrolador.

Una de las variables principales que se deben estudiar antes de elegir algún tipo de transmisión es la cantidad de canales necesarios. En los vehículos que solo se mueven

en un plano horizontal, se necesitaran como mínimo dos canales para modular la aceleración y giro. En caso de que se quiera añadir un servo más como brazo, serán necesarios dos canales más para su movimiento.

2.1.2. Control por Infrarrojos

Otro tipo de comunicación es por el método de infrarrojos (IR). Al igual que el anterior, tiene un emisor y un receptor. En este caso, el emisor desprenderá luz infrarroja que el receptor será capaz de leer a partir de un sensor de infrarrojos y, finalmente, se trasmitirá una señal eléctrica al circuito.

Dependiendo de la comunicación entre ambos, se dividen en enlaces de línea de vista y enlaces sin línea de vista. En los enlaces de línea de vista, la luz llega de manera directa desde el emisor al receptor. El caso contrario, se refiere a que la luz infrarroja emitida es reflejada en alguna superficie antes de llegar al receptor[8].

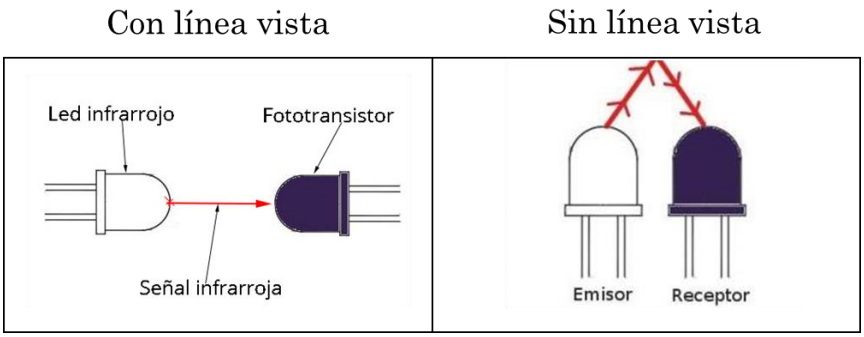


Figura 4. Enlaces en la comunicación por infrarrojos.

Los enlaces sin línea vista se suelen emplear como sistema de detección de obstáculos.

2.1.3. Robots con control remoto integrado

Para este proyecto se empleará un robot previo que ya tiene incluido el control remoto al que se le añadirán ciertas cualidades para que logre los objetivos comentados. Para la elección de este, se busca un dispositivo capaz de ser manejado a distancia que contenga la mayor cantidad de sensores posibles para favorecer el montaje del futuro robot. También se buscará que los precios sean asequibles.

Se han tenido en cuenta dos posibles candidatos como punto de partida. Se estudiaron el *Edison V2.0* y el *mBot* de la empresa Makeblock.

Edison V2.0 STEM Robot

El robot *Edison*[9] se emplea como robot educativo en muchos colegios ya que se considera fácil de programar. Además, es compatible con LEGO en caso de que se le quiera añadir algún elemento externo.

En caso de utilizarlo en el proyecto, se utilizaría la aplicación *EdPy* que utiliza el lenguaje Python. Este software se encuentra online de manera gratuita. El robot tiene 5kB de memoria para poder descargarse grandes programas y un procesador *Freescale 8-bit MC9S08PA8VLC*.

El robot tiene la capacidad de conectarse a cualquier mando a distancia y un módulo receptor de infrarrojos de 38kHz. Además, una ventaja respecto a modelos anteriores es que tiene codificadores en cada una de las ruedas para poder definir la velocidad y dirección exacta a la que se quiere controlar al robot.

El robot está diseñado para detectar obstáculos, seguir líneas, detectar la luz y seguir instrucciones bajo control remoto. Todo esto es posible gracias al hardware instalado:

- Sensor de sonido (transductor piezoeléctrico).
- Altavoz piezoeléctrico.
- Dos botones programables.

- Dos LEDs rojos en la parte frontal del robot.
- Dos LEDs infrarrojos en el frontal.
- Dos fototransistores en la parte frontal del robot y otro en la parte inferior.
- Módulo receptor de infrarrojos en la parte frontal del robot.

Figura 5. Mapa de sensores de Edison V2.0.

En cuanto al sistema de alimentación, únicamente son necesarias 4 pilas triple A. En caso de que el robot se encuentre en un tiempo de inactividad mayor de 5 minutos, se cambiará a modo reposo para ahorrar energía. Las dimensiones físicas del robot se encuentran en la siguiente tabla:

Dimensiones	7,5x4x8,5 cm
Peso	249 gramos
Precio	50 €

Tabla 1. Características físicas de Robot Edison V2.0.

MBOT Makeblock

El *mBot* de Makeblock[10] también es un robot creado para niños. Este también tiene integradas ciertas funciones de fábrica que se podrán aprovechar. Entre ellas, evitar obstáculos, seguimiento de líneas y control manual.

Para su programación se puede usar el programa *mBlocks* y *Arduino IDE*. A diferencia del Edison, el *mBot* tiene una placa *mCore* accesible para poder conectar más dispositivos en caso de que fuese necesario.

En diferencia al robot *Edison*, el *mBot* tiene conexión bluetooth integrada. Además, también tiene los siguientes sensores[11]:

- Sensor de luminosidad en la placa.
- Emisor y receptor de infrarrojos en la placa.
- Dos luces LED
- Zumbador
- Pulsador programable
- Sensores ultrasonido en la parte frontal. Mide distancias de 3 centímetros a 4 metros en un ángulo frontal de 30º de apretura.
- Sensor sigue-líneas en la parte inferior del robot. Tiene dos emisores receptores de infrarrojos.

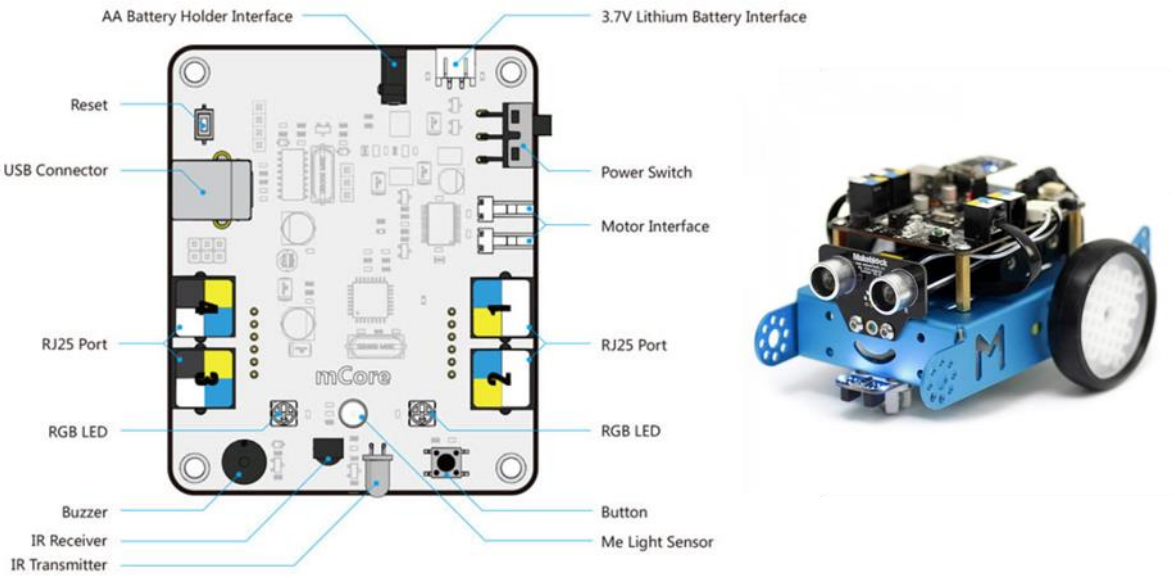


Figura 6. Mapa de sensores de mBot de Makeblock.

Otra de las ventajas del *mBot* es que ofrece diferentes opciones a elegir para la alimentación: entre cable USB, 4 pilas AA o una batería de litio 3.7 V. Las características físicas en este caso:

Dimensiones	17x13x9 cm
Peso	500 gramos
Precio	99 €

Tabla 2. Características físicas de mBot Robot.

2.2. Servidor web

Un servidor web, o también llamado servidor HTTP, sirve para almacenar archivos que podrán ser alcanzados desde el navegador de cualquier cliente. En otras palabras, un servidor web responde a las peticiones del servidor de un cliente, ejecutando los archivos que almacena y mandando la información a través de direcciones IP.

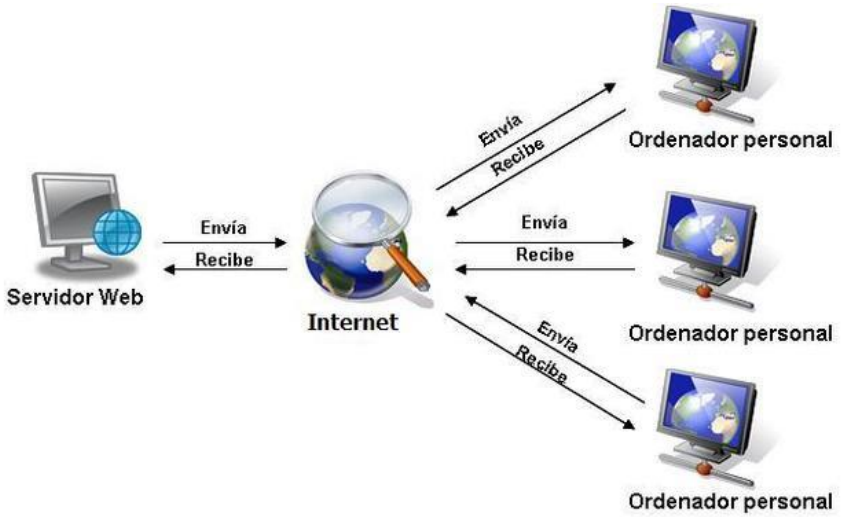


Figura 7. Esquema de un servidor web.

Por otro lado, el protocolo HTTP o protocolo de transferencia de hipertexto es el tipo de comunicación que mueve la información en un método de petición-respuesta. Este protocolo es unidireccional, y el servidor dará una respuesta a las peticiones del cliente. Una vez el servidor haya dado una respuesta a la petición del cliente, se cerrará el hilo de comunicación a no ser que se le haya especificado al servidor que no lo haga.

En el caso de un servidor *streaming*, se le ha especificado al servidor HTTP que tiene que mantener el canal de respuesta abierto, aunque se haya dado una respuesta al cliente. Se suele usar para la transmisión de vídeos o audios en vivo.

El protocolo TCP o Protocolo de Control de Transmisión, es uno de los protocolos de transporte que utilizan los HTTP. Se utiliza para asegurar que los datos se transmiten en orden y que llegan al destino sin errores. También se encarga de controlar que la red no se sature controlando el flujo de datos en la comunicación. Los puertos en los protocolos TCP sirven para distinguir las puertas emisoras y receptoras en la comunicación. En un servidor HTTP, el puerto para un servidor web es el número 80.

Un protocolo alternativo al HTTP, son los llamados *WebSockets*[12]. A diferencia del HTTP, mantiene una comunicación *full-duplex* entre servidor y cliente. Esto resulta una ventaja en cuanto a velocidad en webs a tiempo real ya que con este método se abre un canal bidireccional que se mantendrá abierto hasta que uno de los dos elementos decida cerrar la comunicación. En este protocolo el cliente manda una petición para conectarse y automáticamente recibe como respuesta un *HandShake* o permiso de conexión por parte del servidor. Seguidamente, se abre el hilo de comunicación entre cliente y servidor.

2.3. Comunicación por imagen

Otro de los componentes esenciales para el proyecto es la comunicación visual. Se quiere introducir una cámara en el robot que se comporte como los ojos del usuario. Así, la imagen se podrá alterar para simular daltonismo y ceguera. Por tanto, se necesita una cámara capaz de mostrar la imagen en vivo con la posibilidad de alterar el color de la imagen. En caso de que esto último sea muy complicado, se añadirá un servomotor paso a paso que bajará una lente que distorsione el color físicamente.

Para la transmisión de imagen existen dos posibilidades: conexión Wifi o conexión Bluetooth. A diferencia de una red Wifi, una red Bluetooth únicamente puede conectar dispositivos en pares. Una red Wifi podría conectar a más de dos dispositivos a la vez. Además, una desventaja de una conexión Bluetooth es la necesidad de una distancia relativamente corta para mantener la comunicación frente a la que necesita una red Wifi.

A priori, por lo que se ha expuesto anteriormente, ambas propuestas son válidas para lo que se pretende en este proyecto. Sin embargo, una red Wifi limitaría menos en

caso de que se quieran añadir más aplicaciones a este proyecto en un futuro. El robot podrá ser manejado desde grandes distancias siempre que esté conectado a la misma red Wifi, y si se desea, se podrá conectar a otros dispositivos a parte del control remoto del usuario.

Teniendo en cuenta la decisión anterior, se plantearon las siguientes cámaras inalámbricas para el proyecto:

- La Cámara *Raspberry Pi Camera Module V213* es el módulo que ofrece la compañía Raspberry. Tiene un sensor de 8 megapíxeles *Sony IMX219*. Este accesorio puede hacer fotografías y grabar vídeos que se reproducen a tiempo real en otra pantalla si se desea. Ofrece la posibilidad de alterar la imagen y crear efectos. El inconveniente es que sería necesario comprar una placa que permita ese tipo de puerto de entrada, lo que elevaría el precio del conjunto. Tiene muchas opciones de edición gracias a *Picamera*, una librería de Python. El precio, únicamente de la cámara, es de 25 €.

Figura 8. Raspberry Pi Camera Module V2.

- La mejor opción es posiblemente el conjunto *ESP32-CAM*[14] con la cámara *OV2640*[15]. La placa contiene una salida *PWM* que se podrá aprovechar para mover el brazo del robot. Tiene conexión Wifi pudiéndose conectar al ordenador y al móvil. Además, se le puede alterar el color de imagen en tiempo real desde el ordenador. El único inconveniente es que no tienen ninguna entrada para poder conectarlo al ordenador. Será necesario comprar también un convertidor *FTDI FT232RL* (4 €). Esta placa se puede programar con Arduino IDE. Se puede encontrar todo el conjunto por menos de 20 €.

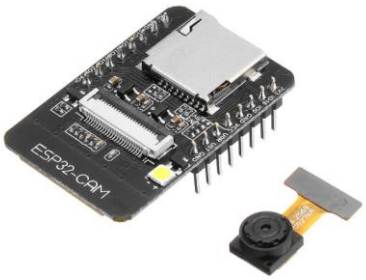


Figura 9. Modulo ESP32-CAM y cámara OV2640.

- La cámara de vigilancia *YI*[16] tiene una resolución de 1080P. Es una cámara Wifi que permite la conexión al teléfono móvil para la visualización de la imagen a tiempo real. El inconveniente de esta cámara es que no permite la distorsión de imagen, por lo que será necesario añadir un servomotor paso a paso para dejar caer una lente que distorsione el color físicamente. El precio de la cámara es 25 euros.

Figura 10. Cámara de vigilancia YI.

- Otra opción parecida a la anterior es la cámara *Xiaomi IP Xiaofang*[17]. Tiene también una resolución de 1080P y al igual que la anterior tiene conexión Wifi pudiéndose conectar al teléfono móvil. Tiene la ventaja de que se puede hablar a través del teléfono móvil y que se escuche a través del altavoz de la cámara, lo que podría ser útil para futuras aplicaciones. El precio en el mercado es inferior a los 25€.

Figura 11. Xiaomi IP Camera Xiaofang.

- La cámara V380[18] es también parecida a estas últimas dos. Tiene la misma resolución que las cámaras anteriores y tiene una aplicación para ver la imagen a tiempo real por medio de Wifi. El precio es inferior a $20.

Figura 12. Cámara V380.

2.4. Robot Zuri

Desde 2014, la empresa alemana *Zoobotics* trabaja en un nuevo prototipo llamado *Zuri*[19]. Se compone en su mayoría por papel y cartón gris, que son materiales reciclables y biodegradables, lo que estaría en consonancia con el objetivo medioambiental del proyecto. Es programable y ofrece control remoto por bluetooth.

Al igual que el robot *Zuri*, el robot de este proyecto intentará componerse en su mayoría de productos reciclables y ecológicos.

Figura 13. Robot Zuri de Zoobotics.

2.5. Brazo robot MeArm

El robot MeArm[20] es un proyecto que comenzó en 2014 y ha presentado su última versión en 2018. Se trata de un brazo robot tamaño bolsillo que simula brazos robot de tipo industrial. Se puede controlar a partir de las salidas PWM de un microcontrolador Arduino o un Raspberry Pi. Aunque se ofrezcan varias versiones, todas son muy parecidas. Las articulaciones son servomotores, y la mano es sustituida por una pinza. El brazo del proyecto estará basado en esta idea.

Excepto por una versión, todos los brazos robot de esta compañía están fabricados en plástico. Esta excepción fue fabricada a partir de MDF o fibras de densidad media. Se pensó en este material como sustituto más ecológico al plástico, ya que las fibras de madera son residuos de madera aprovechados. Por otro lado, el formaldehido que compone el pegamento evita que este material se pueda reciclar, dejándolo destinado a la incineración[21] tras un único uso.

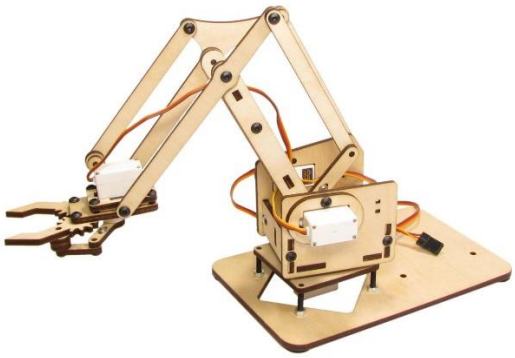


Figura 14. MeArm de MDF.

Existe la posibilidad de que el MDF se pueda usar para el proyecto en caso de que se sustituya el formaldehido por una resina creada a partir del almidón de la patata. Esto último todavía es sujeto de estudio[22].

3. Arquitectura del Proyecto

En este capítulo se expondrán los elementos necesarios para llevar a cabo el proyecto, además de una explicación de por qué se han seleccionado específicamente. También se expondrán los recursos utilizados. En los capítulos posteriores se explicará su funcionalidad.

Todo el robot se podría dividir en dos subsistemas que funcionan de manera completamente independiente. El primero se compone únicamente por el robot Edison Robot V2.0 que funciona como plataforma y estará controlado a partir de un mando de infrarrojos. Al estar alimentado por pilas, su fuente de alimentación será independiente al resto de la estructura. En segundo lugar, la placa ESP32-CAM que será la única placa del sistema. En ella estará conectada el resto de los elementos del robot: cámara, servomotores y fuente de alimentación adicional. Esta se conectará por Wifi a cualquier dispositivo por medio de un servidor web que ofrece la placa.

Por último, el brazo robot de la estructura, estará unido a la plataforma Edison, pero estará a su vez conectado a la placa a través de los servomotores.

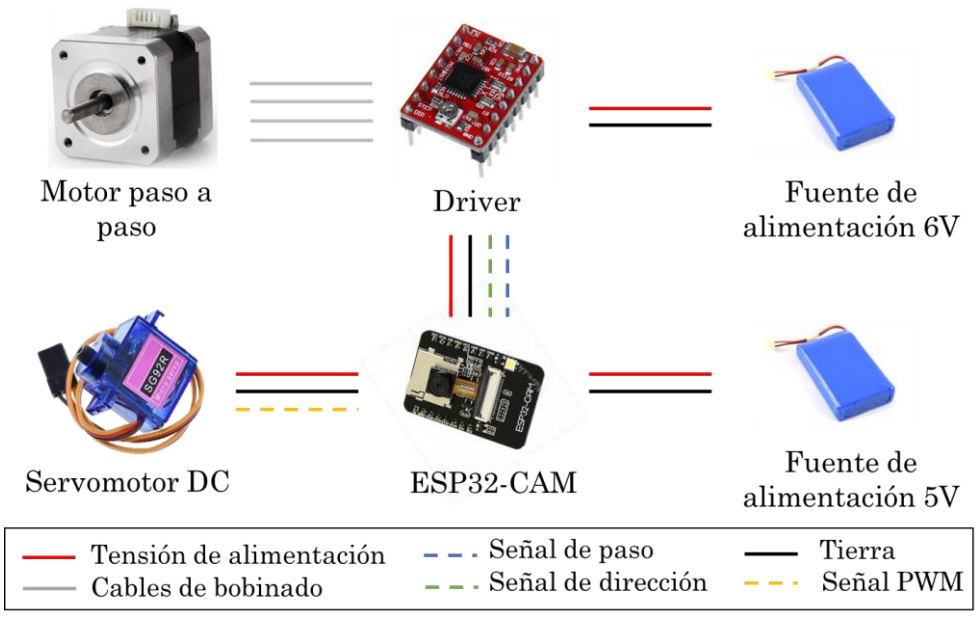


Figura 15. Conexionado de ESP32-CAM.

3.1. Vehículo Edison V2.0 STEM

A diferencia del *Edison*, el *mBot* tiene pines accesibles en la placa *mCore* para añadir cualquier otro tipo de componente electrónico. Pero por la diferencia de precio entre ambos robots, existe la posibilidad de que resulte más factible complementar el *Edison* con otra placa externa independiente, ya que, definitivamente serán necesarios unos pines accesibles para la conexión del servomotor y de la cámara.

Para las funciones que se esperan del robot, en principio sólo son necesarios los sensores ultrasonido o infrarrojos para evitar los obstáculos. Puesto que estos sensores están incluidos en ambos, ambos son válidos y escoger el robot *Edison* frente al *mBot* no supondría ninguna desventaja.

Además, la propia compañía ofrece ciertas aplicaciones gratuitas para la programación del robot *Edison*. Estas aplicaciones permiten escribir el código y subirlo al robot desde una página web sin necesidad de licencias o descargas en el ordenador. Una de las aplicaciones que se ofrecen es *EdPy*, que utiliza el lenguaje Python.

Por estas razones, se ha decidido emplear el robot *Edison V2.0 STEM*. Puesto que ambas opciones eran viables y muy parecidas, la decisión se ha tomado en función al precio y a la accesibilidad, ya que la universidad ya está familiarizada con este robot.

3.2. Módulo ESP32-CAM y Cámara OV2640

Finalmente, para la cámara inalámbrica se va a emplear el conjunto *ESP32-CAM* y cámara *OV2640*, y toda la programación relativa a la placa se hará desde Arduino IDE debido a las facilidades que ofrece para crear un servidor web simplificado.

Es posible mantener una comunicación inalámbrica con la placa desde cualquier dispositivo con acceso a internet gracias a su conectividad wifi 802.11 compatible con estándares b, g y n, además de ofrecer conexión bluetooth v4. 2. También es compatible con los modos de operación STA/AP/SAT+AP. La placa posee el chip ESP32-S con un microprocesador CPU de 32-bit de doble núcleo con un rendimiento de 600 DMIPS y velocidad de 160MHz. Tiene una memoria SRAM de 520 KB y es compatible con UART, SPI, I2C, PWM, ADC y DAC[23].

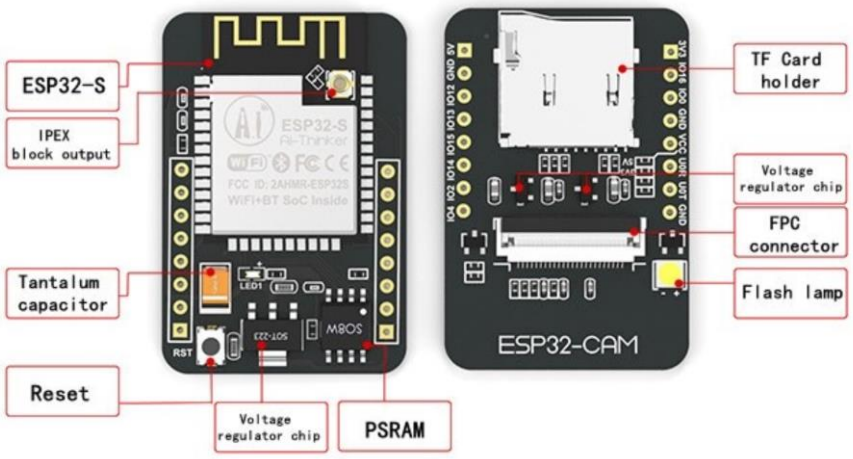


Figura 16. Componentes de la ESP32-CAM.

La imagen puede ser retrasmitida en vivo desde un servidor *streaming*. Además, se podría mantener una comunicación bidireccional, en la cual, el usuario tras del dispositivo pueda controlar los pines GPIO de la placa. En resumen, este conjunto responde a todas las necesidades previstas para el proyecto, siendo además la posibilidad más económica.

En principio, la placa, que también tiene un LED integrado a modo de flash si se necesita, está conectada al sensor de imagen *OV2640*. En caso de que se prefiera una cámara con mayor resolución bastará con sustituirlo por la versión *OV7670*. La razón por la que se ha decidido mantener la *OV2640* es únicamente por facilidad y porque no se ha considerado que para la finalidad de la cámara se necesite mayor calidad que los 2 megapíxeles que tiene la elegida. Otra característica para tener en cuenta es que el ángulo de incidencia del sensor de imagen es de 25º de apertura[24].

3.2.1. Pines GPIO de la ESP32-CAM

En cuanto a los pines GPIO, algunos de ellos tienen funciones predeterminadas. Existen tres pines GND y dos pines de voltaje para 3,3V o 5V específicamente. Además, el pin VCC ofrece el voltaje al que está alimentada la placa como salida. Los pines GPIO 1 y 3, son pines seriales, que se utilizarán para la transmisión de datos entre la placa y el ordenador cuando sea necesario. El pin 3 está etiquetado como recibidor de datos, y

el 1 como transmisor. El GPIO 16 es por defecto un pin UART. Por último, la placa únicamente podrá cargar el código si el pin 0 está conectado a tierra.

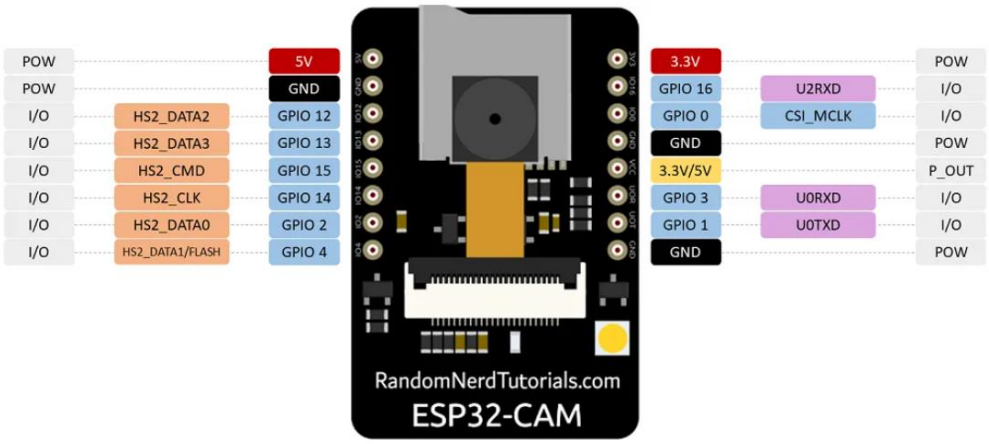


Figura 17. Pines GPIO de la ESP32-CAM.

Los pines 2, 4, 12, 13, 14 y 15 soportan entradas analógicas, y son los pines a los que se conectaran el servomotor y el motor paso a paso.

3.2.2. Servidor Web en la ESP32-CAM

Este producto facilita la manera de realizar un servidor web desde Arduino IDE[25], que ya ofrece ciertos ejemplos para la programación de las tarjetas ESP32. En uno de estos ejemplos, existe un código denominado '*CameraWebServer*' que crea un servidor *streaming* con el que se puede retransmitir la imagen en vivo y una serie de elementos que permiten alterar la imagen trasmitida. Este código se usará como base, para la realización del código final.

En primer lugar, se conectará la placa a una red wifi. Una vez conectada, la placa tendrá accesible una dirección IP a la que se podrán conectar los dispositivos de los clientes.

Cada vez que un dispositivo nuevo se conecte a estas direcciones IP, la placa recibirá una llamada petición HTTP. En respuesta, la placa devolverá el documento HTML que tendrá incluido en su código para que el cliente pueda visualizar el contenido. La

comunicación se mantendrá abierta vía internet, permitiendo al usuario controlar los pines GPIO de la placa y la imagen emitida por el servidor *streaming* en la placa.

Como requisito para esta comunicación, la placa debe estar conectada a una red wifi durante todo el proceso. Por otro lado, no es necesario que los usuarios este conectados a la misma red wifi con tal de que tengan acceso a internet.

3.2.3. Convertidor USB a Serial TTL

Debido a que la placa ESP32-CAM no tiene ningún puerto para poder conectarlo al ordenador y así programarlo, es necesario añadir un nuevo elemento al proyecto. Se necesitará un adaptador USB a Serial TTL (Transistor-Transistor-Logic) para la comunicación entre ordenador y placa en el momento de la programación.

Como cualquier otro convertidor, funcionará como interlocutor entre la placa y el puerto USB del ordenador. Para el proyecto se ha empleado el convertidor *FT232RL* de la empresa FTDI, pero cualquier otro convertidor que tenga las entradas o salidas parecidas a este, sería válido. Como se puede apreciar en la siguiente ilustración, tiene acceso a las señales GND, CTS (Clear To Send), VCC, TX (transmisor de datos), RX (receptor de datos), y DTR (Data Terminal Ready).

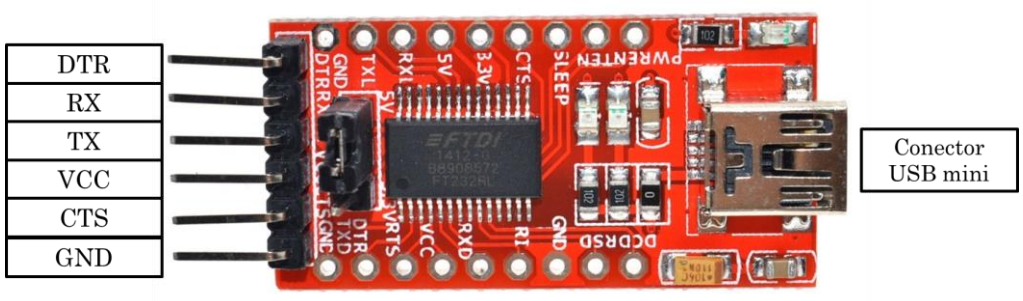


Figura 18. Mapa de convertidor USD-Serial TTL ft232RL de FTDI.

Puesto que el conector es USB mini, es también necesario un cable USB mini a puerto USB normal.

3.3. Comunicaciones

En la siguiente figura se esquematiza el sistema de comunicación que siguen los elementos.

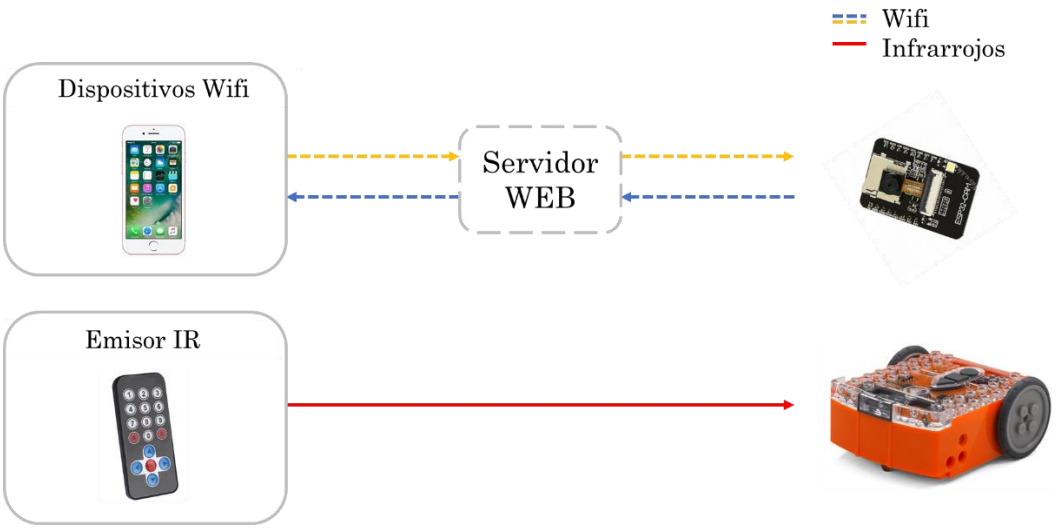


Figura 19. Esquema de comunicaciones.

En primer lugar, el Edison V2.0 que funciona como la plataforma del robot, será controlado por un mando a distancia que emita infrarrojos. Este guiará su movimiento por el plano horizontal permitiéndole avanzar en las cuatro direcciones.

Por otro lado, y de manera completamente independiente, se encuentra el control de la placa. Esta placa se controlará vía Wifi a partir de cualquier dispositivo. Este dispositivo debe ser capaz de conectarse al servidor web que ofrece la placa ESP32-CAM. La comunicación en este caso será bidireccional, ya que la imagen será retrasmitida en el dispositivo, al mismo tiempo que el usuario será capaz de mover el brazo desde el mismo dispositivo. Todo esto se puede hacer gracias al servidor web.

La ventaja de que las dos comunicaciones sean independientes es la posibilidad de sustituir el robot Edison que sirve de plataforma de manera sencilla por otro robot distinto.

4. Elección del material

Uno de los objetivos de este proyecto es logra la menor incidencia ambiental. Por eso se procurará que el material del robot sea reciclado, fácilmente reciclable o biodegradable.

4.1.1. Aluminio

El aluminio es un material que puede ser reciclado en su totalidad sin perder calidad, prolongando infinitamente su vida útil. Una de las cualidades más sorprendentes de este material es que la energía necesaria para reciclar el material es un 5% de la energía necesaria para su obtención primaria. Siendo mucho más rentable reciclar el material antes de conseguirlo de primera mano.

La densidad de este material es de aproximadamente 2,7 g/cm^3. Su resistencia es un tercio a la resistencia del acero. Tiene un módulo elástico de 65 GPa pero se puede someter a procesos térmicos para mejoras sus propiedades físicas.

4.1.2. Cartón gris

Se ha mencionado con anterioridad el robot *Zuri*, que emplea el cartón gris y el papel como materiales principales, reduciendo el coste de materia prima del robot y reduciendo su impacto al medio ambiente. El cartón gris se produce a partir de papel reciclado, y los procesos químicos que se emplean para su fabricación no comprometen su posterior reciclado. Además, es 100% biodegradable.

Figura 20. Cartón gris.

El cartón idóneo para este trabajo sería el que ofrezca la mejor rigidez a flexión con el mínimo gramaje. Los gramajes en la actualidad están comprendidos entre los 300 gsm y los 3000 gsm[26]. Por ello, se elegirá el tipo de cartón atendiendo a la rigidez y peso que aporte. La rigidez del material viene dada según la norma *Taber ISO* 2493, que mide la fuerza necesaria para doblar la plancha 5º o 15º. Para el estudio de la rigidez a flexión se debe tener en cuenta que dependiendo de la dirección en la que se aplique la fuerza se obtendrán resultados diferentes, ya que el cartón es un material anisótropo al tener las fibras colocadas en una dirección concreta. Dependiendo de la calidad del cartón, se le clasifica en diferentes grados.

Grade	Description	Gramme	Thickness
B	One layer board, two side grey	750g~1500g	1.24~2.55mm
A	One layer board, two side grey	320g~670g	0.49~1.07mm
A	Laminated board, two side grey	720g~2400g	1.15~4.0mm
AA	One layer board, full grey	370g~670g	0.57~1.05mm
AA	Laminated board, full grey	750g~2150g	1.13~3.4mm
AAA	Laminated board, glossy surface	950g~1900g	1.27~2.9mm

Figura 21. Clasificación del cartón gris por calidades.

Según la compañía *RDM Group*, la rigidez de este material con un grosor de 1,8mm y un gramaje de 850 g/m^2, puede llegar a valer 161,9 mN·m bajo la norma *Taber*[27]. Sigue siendo una rigidez pequeña comparada a lo requerido por la estructura de este proyecto, pero al tratarse de un grosor tan pequeño, se podrían sobreponer varias capas.

Otro ejemplo es el cartón gris que ofrece la empresa *MM Karton* bajo el nombre de *GK 33 Greyboard*. Ofrecen un cartón de 700 g/m^2 de gramaje con un grosor de 1mm y una rigidez longitudinal a los 15º de 110 mN·m según la norma Taber[28].

Por último, la compañía *New Bamboo Paper*, ofrece un cartón gris de entre 2mm y 5 mm de grosor con un gramaje de entre 2000gsm y 5000gsm. Aunque no enseñan la resistencia mecánica del material, la empresa dice emplear este cartón para muebles[29].

5. Diseño mecánico

En este capítulo se explicará el diseño mecánico pensado para el brazo del robot. Se explicarán las decisiones tomadas y se presentaran los cálculos que justifican dichas decisiones. Además, se especificarán los componentes electrónicos y mecánicos necesarios para esta parte del proyecto.

El conjunto del brazo mecánico estará formado de tres partes que se nombrarán de la siguiente manera: base, antebrazo y pinza. La metodología que se seguirá para diseñar el conjunto es empezar desde la pinza hasta la base, ya que no se puede diseñar una base sin haber especificado con anterioridad el peso que debe resistir.

Hay que añadir que el brazo robótico no tendrá muñeca, lo que significa que no podrá coger objetos desde todos los ángulos, sino únicamente desde un plano horizontal. Esto se ha decidido así para simplificar el proyecto, ya que la finalidad del trabajo es hacer un robot completo que cumpla otras expectativas.

La base será el método de unión entre el antebrazo y el resto del robot. Funcionará a modo de codo u hombro, permitiendo un movimiento en el plano XZ a partir del impulso del primer servomotor de la estructura. Estará colocada en la superficie superior del robot Edison V2.0. Esta base contendrá el módulo de cámara ESP32-CAM junto con la fuente de alimentación, además de ser el medio de unión entre el brazo mecánico y el resto del conjunto.

Puesto que la extremidad robótica no constará de muñeca, uno de los requisitos del antebrazo será incluir un sistema que permita que la pinza traslade al elevarse el antebrazo, manteniéndose siempre paralela al suelo. Así, se podrán recoger objetos desde cualquier altura. También deberá tener en cuenta el ángulo de apertura de la cámara para que la pinza este siempre en el campo visual de la imagen.

Por último, la pinza simulará una mano que tendrá la fuerza suficiente para levantar una carga máxima de 250 gramos. Para su funcionamiento también será necesario incluir un servomotor que estará conectado a su vez a la placa ESP32-CAM.

La pinza estará unida al extremo del antebrazo. Como se ha expresado anteriormente, la pinza fue el primer paso en el diseño del conjunto.

5.1. Pinza

Para el diseño de la pinza no se tenía ninguna especificación, ya que el entorno interactivo en el que se pretende introducir al robot es fácilmente alterable en función a lo que pueda resistir. Aun así, se han querido especificar unos límites iniciales para llevar a cabo el diseño. Estos han sido que el objeto a levantar pueda tener unas características máximas de 4 o 5 centímetros de ancho y un peso de 250 gramos.

Para el diseño se han ordenado los segmentos en forma de paralelogramo articulado[30]. De esta manera, existe traslación en el dedo de la pinza alrededor de la muñeca. En la siguiente imagen se puede apreciar como los segmentos opuestos del mecanismo están en paralelo durante todo el recorrido. Con la traslación del segmento superior se asegura que la pinza podrá sostener cualquier objeto, ya que las caras internas de la pinza son siempre paralelas al eje de simetría del conjunto y, por tanto, entre ellas.

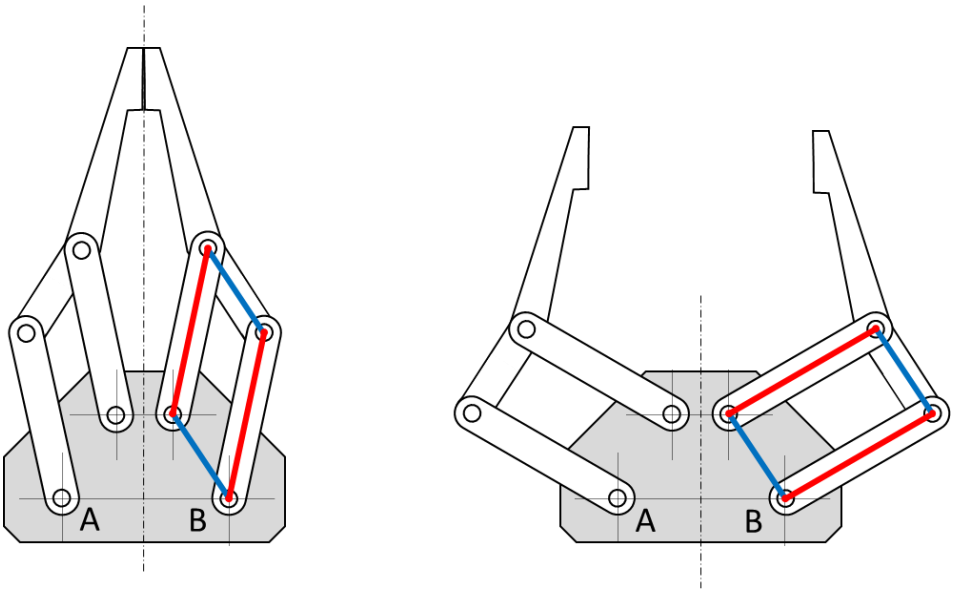


Figura 22. Diseño mecánico de la pinza I.

El mecanismo será empujado por un servomotor que transmitirá la potencia a un sistema de engranajes. Estos engranajes estarán conectados a los puntos de articulación A y B, forzando al mecanismo a moverse simétricamente. El servomotor transmitirá su potencia a uno de los engranajes que, a su vez, la transmitirá al siguiente.

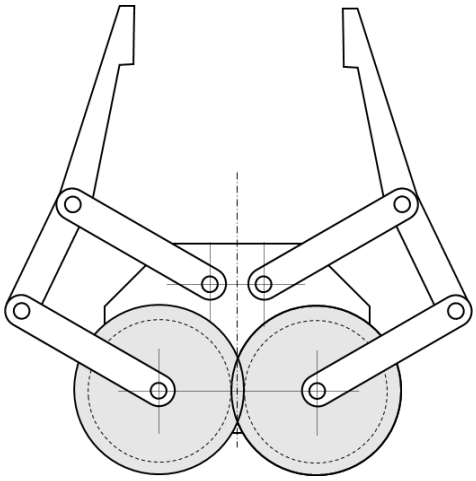


Figura 23. Diseño mecánico de la pinza II.

Una vez decidido el movimiento general del mecanismo, se realizaron los cálculos necesarios para elegir engranajes y la potencia del micro servo para asegurar un correcto funcionamiento.

En principio el rozamiento entre los segmentos de la pinza se considerará despreciable para la parte de diseño. Más adelante, se estudiará en las pruebas el efecto de dicha fuerza en el funcionamiento de la estructura y una posible solución para enfrentarla.

5.1.1. Estudio estático de la pinza mecánica

Para especificar las características mecánicas de la pinza se necesita comprobar el comportamiento que tendría la pinza sujetando un objeto. Para ello se hará un estudio estático en el momento en el que la pinza sostiene el objeto en el aire. El siguiente esquema es un diagrama de fuerzas sobre la carga que es agarrado por la pinza. En él

se refleja el peso del objeto ($m \cdot g$), las fuerzas de rozamiento entre la superficie del objeto y los dedos ($\mu \cdot N_1, \mu \cdot N_2$), y la fuerza de pinzado que ejerce el robot sobre el objeto ($F_{pinzado}$).

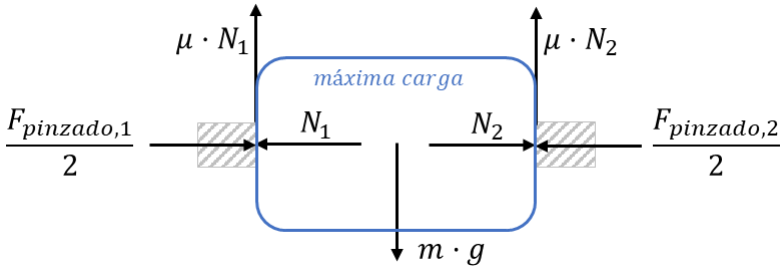


Figura 24. Estudio estático de la pinza I.

Suponiendo la figura estática, se puede calcular la relación entre todas estas fuerzas a partir de la Ley de Newton que dice que el sumatorio de todas las fuerzas externas ejercidas sobre un mismo objeto estático es nulo. En la figura anterior las fuerzas ejercidas sobre cada eje deben ser cero, resultando que las fuerzas normales son iguales en módulo y, por consiguiente, las fuerzas de rozamiento son totalmente simétricas respecto al eje Y. A partir del sumatorio de fuerzas sobre el eje vertical se puede llegar a la siguiente fórmula, que relaciona la fuerza de pinzado con el peso de la carga y el coeficiente de rozamiento μ. Puesto que el peso de la carga y el coeficiente de rozamiento se suponen conocidos, la fuerza de pinzado máxima que debe ejercer cada uno de los dedos de la garra para levantar un objeto es la siguiente.

$$F^{MAX}_{pinzado\ por\ dedo} = \frac{m_{MAX} \cdot g}{2 \cdot \mu}$$

El siguiente esquema simula de manera simplificada la mitad izquierda del cuerpo de la pinza. Este diagrama de fuerzas tiene representadas el par del servo (M), la fuerza que genera este par en el segmento ABC o dedo (F_M) y por último la fuerza de pinzado mencionada anteriormente ($F_{pinzado}/2$).

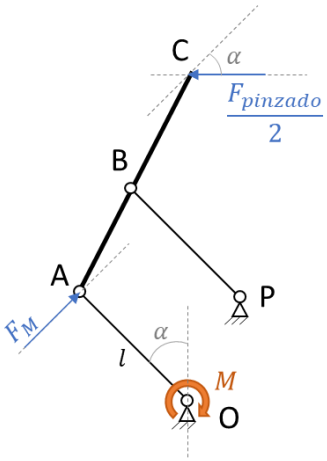


Figura 25. Estudio estático de pinza II.

Aplicando la ecuación del momento y la Ley de Newton sobre el segmento ABC, se obtiene la siguiente relación. Además, aplicando la ecuación obtenida anteriormente sobre la fuerza de pinzado, se concluye que el par motor debe cumplir siempre lo siguiente:

$$F_M = \frac{F_{pinzado}}{2} \cdot \cos(\alpha) \rightarrow M = l \cdot \frac{F_{pinzado}}{2} \cdot \cos(\alpha)$$

$$M = \frac{m \cdot g \cdot l}{2 \cdot \mu} \cdot \cos(\alpha)$$

En conclusión, el momento que hay que aplicar a la pinza para lograr un correcto funcionamiento depende de la carga, de la longitud de los segmentos paralelos OA y PB, del coeficiente de rozamiento entre garra y carga y del ángulo de apertura de la garra en cada momento. A partir de esta fórmula se pueden sacar las siguientes conclusiones:

- El instante crítico en el comportamiento de la pinza es cuando el ángulo α es mínimo, cuando la pinza está cerrada. En este momento, la fuerza de pinzado queda neutralizada únicamente por la fuerza ejercida por el par. En los casos en los que el ángulo de apertura α es mayor, el momento solo debe superar a las proyecciones de la fuerza de presión sobre su eje, siendo estas más pequeñas que la fuerza total. Consecuentemente, el par máximo será cuando el ángulo sea menor.

- Al estar el coeficiente de rozamiento en el denominador, el mecanismo será más eficaz cuanto mayor sea este coeficiente. En caso de que el material elegido no tenga un coeficiente lo suficientemente alto, se podrá adherir a los dedos del mecanismo una capa de goma que asegure la sujeción del objeto. Para simplificar el proceso se buscará que el coeficiente de rozamiento entre las superficies de ambos materiales sea elevado.
- La única medida geométrica de la que depende directamente la fuerza del mecanismo es la longitud de los segmentos OA y PB. Esta medida deberá ser lo más pequeña posible para que el par requerido por el motor no sea excesivamente alto. Sin embargo, esta decisión debe tener en cuenta, no solo la magnitud del par, sino la geometría de la pinza, ya que disminuir excesivamente esta longitud podría ocasionar otros problemas como que la distancia de apertura se vea muy disminuida.

Por las ideas anteriores se podría concluir que se debe buscar la manera de aumentar el valor de α y disminuir la longitud de los segmentos OA y PB. Sin embargo, esto no es posible ya que, al aplicar las reglas trigonométricas, se llega a la conclusión de que al aumentar uno, aumenta el otro. Por eso, es necesario decidir cuál de los dos valores tiene una mayor influencia en el resultado. El coseno del ángulo es menos influyente que la variación de la longitud de los segmentos paralelos.

5.1.2. Elección de componentes de la pinza

Para finalmente definir el valor máximo del par, se han hecho diferentes suposiciones. Al estar el material todavía indefinido, el coeficiente de rozamiento para este estudio se ha especificado en 0,35, considerando este valor lo suficientemente bajo para considerar el caso como crítico. Siempre será posible añadir material de goma entre pinza y carga para aumentar este coeficiente. La carga máxima es de 250 gramos, resultando la fuerza de pinzado por dedo de 3,5N y una fuerza de pinzado resultante de 7N.

Se busca que la longitud de apertura de la pinza este alrededor de los 5 centímetros. Se intentará que sea lo mayor posible sin comprometer excesivamente la estructura.

Finalmente, se ha decidido que la longitud de los segmentos paralelos debe ser de 2,6 centímetros, que permitirá una distancia máxima de 5,2 centímetros a los 90º de apertura y 5 centímetros al llegar a un ángulo de 77,94º con la vertical. El momento máximo que el servo debe aportar es de 9,098 N-cm o 0,928 kg-cm por dedo de la pinza.

Puesto que el torque necesario para mover una mitad de la pinza es 0,928 kg-cm, el total necesario será el doble, siendo 1,856 kg-cm. Para buscar un motor que mueva dicho mecanismo existen dos posibles opciones: usar dos servomotores con toque suficiente para mover una mitad, o usar un único servomotor que transfiera su potencia a la otra mitad de la pinza a partir de engranajes. Se escogió la segunda opción por buscar que la pinza pese lo mínimo y porque supone un ahorro de energía conectar un único servo en vez de dos.

El servomotor que se ha decidido emplear para esta parte del brazo es el *Micro Servo Digital 9g SG92R* de Tower Pro[31]. Se ha decidido emplear este porque la relación peso-torque se considera la adecuada para este proyecto. El peso del servo es de 9g siendo su torque de 2,5kg-cm a 4,8V. La velocidad máxima de giro del micro servo es de 0,1sec/60º y las dimensiones de 23x12x21.5 milímetros. Al utilizar este servomotor, el factor de seguridad para asegurar el funcionamiento es 1,35.

Figura 26. Micro Servo Digital 9g SG92R.

También se puede estudiar el servomotor MG92B como posible alternativa. Este servo también tendría un funcionamiento válido ya que el torque de este modelo es de 3,1 Kg-cm a 4,8 voltios. Por el contrario, la razón por la que se optó por la primera opción es porque este servo duplica en peso al anterior.

Puesto que la pinza está pensada para abrirse un ángulo inferior a 90º, teniendo la mayoría de los servomotores una rotación de 180º, se puede restringir el ángulo de manera mecánica o bien restringirlo en la programación del robot. Para restringir el movimiento de manera mecánica es necesario añadir un engranaje extra al sistema. Este engranaje será la rueda motriz conectada al servo, que empujará a las dos ruedas que conectan con la pinza. Esta rueda únicamente está pensada para reducir el ángulo de giro del servomotor proporcionalmente. Esta reducción se consigue con la relación de transmisión entre las ruedas dentadas. Siendo el recorrido del servomotor de 180º, y queriendo un ángulo de apertura alrededor de 75º, la relación de transmisión deberá ser 2,4.

En el catálogo de engranajes de RS PRO[32], existen varias combinaciones válidas. Se escogerán las opciones de módulo 0,5 para que el tamaño de la rueda no sea muy grande. Los números de serie de las ruedas elegidas con relación de transmisión 2,4 son DS05-60B y DS05-25B, con 60 y 25 dientes, respectivamente.

Código	Nº dientes	Diámetro primitivo (mm)	Diámetro del agujero (mm)	Diámetro exterior (mm)	Altura (mm)
DS05-25B	25	12,5	4	13,5	7
DS05-60B	60	30	6	31	8

Tabla 3. Medidas de los engranajes en milímetros.

5.2. Antebrazo

El brazo no va a constar de muñeca, siendo imposible controlar remotamente el giro de la mano. Por eso es necesario que el antebrazo tenga incluido un sistema que permita la traslación de la pinza sin rotación alguna. De esta manera, la pinza estará siempre en un plano horizontal pudiendo coger objetos desde cualquier altura. Para esto, se aprovecharán los conocimientos obtenidos en el diseño de la pinza, incluyendo un paralelogramo articulado.

También el movimiento del brazo deberá tener bajo consideración que la pinza, al igual que el objeto a recoger, deben estar comprendidos en el campo visual de la cámara. Como se ha mencionado anteriormente, la cámara con sensor *OV2640* tiene un ángulo de visión de 25º. Si fuese necesario, este ángulo se puede agrandar a partir de una lente. Puesto que el módulo de cámara estará colocado encima de la base, el punto central del objetivo estará situado a 45 milímetros sobre el nivel del suelo. Con el ángulo de apertura de 25º, la imagen llegará a alcanzar el suelo a los 203 milímetros de la ubicación del objetivo.

Para hacer el diseño del antebrazo se estudió el comportamiento del *Robot MeArm V1.0* de Arduino[33]. Al igual que otros brazos MeArm, el antebrazo del robot no es simétrico respecto al plano XZ ya que la mitad derecha de la imagen contiene un paralelogramo articulado adicional que asegura la traslación de la pinza alrededor del eje Y. El antebrazo del proyecto tendrá como ejemplo este robot, aunque incluirá varias distinciones para simplificar el trabajo.

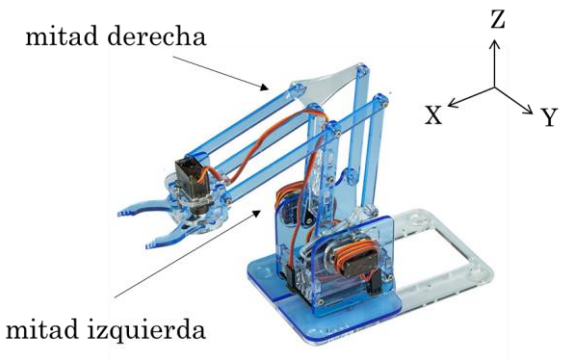


Figura 27. Robot MeArm V1.0 de Arduino.

Una de estas distinciones será la eliminación de las dos mitades del robot, haciendo el antebrazo completamente simétrico respecto al plano XZ. Esta decisión, tendrá como ventaja la optimización eléctrica ya que conlleva la eliminación de un servomotor. Asimismo, se pretende eliminar el servomotor que garantiza la rotación respecto al eje Z, quedando el número de servomotores reducido de cuatro a dos.

A continuación, se presenta el esquema cinemático del modelo propuesto. A diferencia el MeArm, este modelo pretende alargar el segmento OB. El antebrazo estará compuesto únicamente de estos cinco segmentos, que permiten el movimiento del brazo y la traslación de la pinza. Al igual que en el robot MeArm, el segmento OBB' es un segmento fijo, el cual no podrá siquiera rotar alrededor de un eje. Al restringir este movimiento, se asegura que el segmento BB' no se mueve y, en consecuencia, la distancia AA' únicamente se trasladará y se comportará como un segmento rígido. El par rotará el segmento OP, impulsando al segmento ABC a rotar alrededor de B.

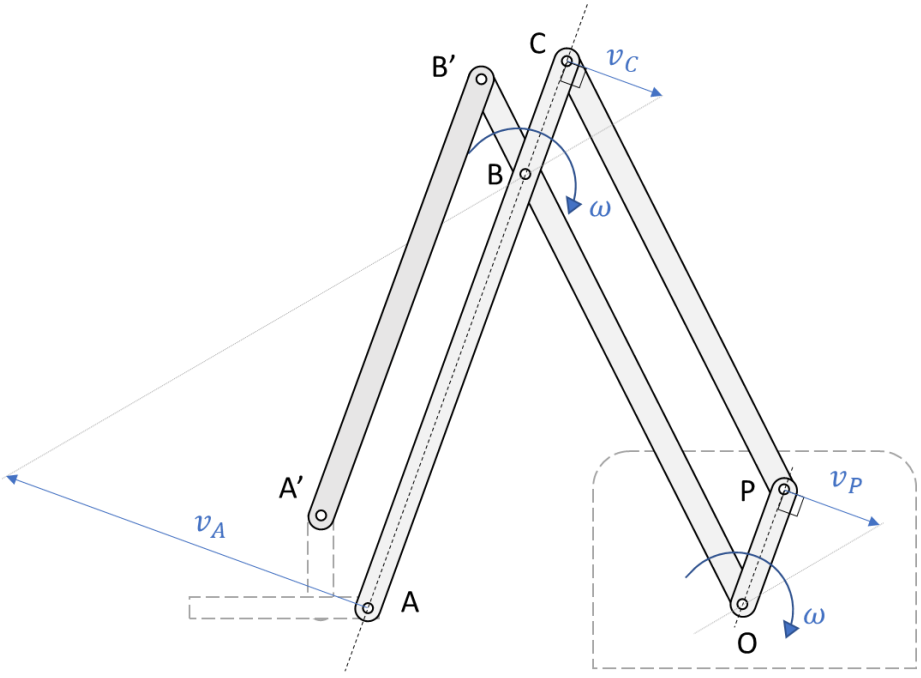


Figura 28. Esquema cinemático del antebrazo.

5.2.1. Estudio estático de antebrazo

Siendo el peso de la estructura desconocido, es difícil calcular la fuerza requerida por el mecanismo. A modo de estimación se puede despreciar el peso de la estructura para calcular los requisitos estáticos. Al hacer un estudio estático ideal se puede concluir que aplicar el par en el punto O, o aplicarlo en el punto B da el mismo resultado, ya que el segmento OBB' se supone fijo y el conjunto OBCP es un paralelogramo articulado. Con eso se concluye que el momento depende principalmente del peso del mecanismo, la longitud del segmento AB y la carga total de la pinza.

Puesto que la inclinación de los segmentos variará dependiendo de la función del antebrazo, se debe estudiar el caso en momento más crítico. Este se considera el punto en el que el segmento OP y, por tanto, el A'B' y ABC, se encuentra en posición completamente horizontal. En ese instante, toda la fuerza del peso del objeto levantado recae sobre el punto A de manera perpendicular al segmento ABC.

En este caso, los 2,45N o 250 gramos que se habían considerado en el apartado anterior, recaen en el punto A en dirección vertical. A estos casi 2,5 N hay que sumarles el peso de la pinza y el del servomotor de 9 gramos. Puesto que la pinza se estima que tendrá un volumen máximo de 15 cm^3, puede llegar a pesar 41 gramos en el caso de que este compuesta de aluminio, y menos en caso de que este compuesta de cartón. El peso total para levantar por el brazo se estima en 300 gramos o 2,95N. Considerando que la distancia AB puede llegar a tener una longitud alrededor de 10 centímetros, el momento crítico ideal que se debe aplicar sobre el punto O es de 29,5 N-cm.

En el caso no ideal, en el que el peso de la estructura no se considera despreciable, el peso de las barras favorece al movimiento siempre y cuando el ángulo que crea la barra OP respecto a la barra fija OB no supere los 90 grados.

En la siguiente figura se puede ver un ejemplo en el que la barra fija está colocada 30 grados respecto a la vertical y la barra OP está en posición completamente vertical, dejando un ángulo de 30 grados de separación entre ambas. En este caso, al estudiar gráficamente la estática de la barra PC, se puede observar como la proyección de la

fuerza resultante en C sobre el eje X es igual a la fuerza ejercida en P. Por tanto, todo el par ejercido en el punto O se aprovecha en el punto B como si se tratase del caso ideal.

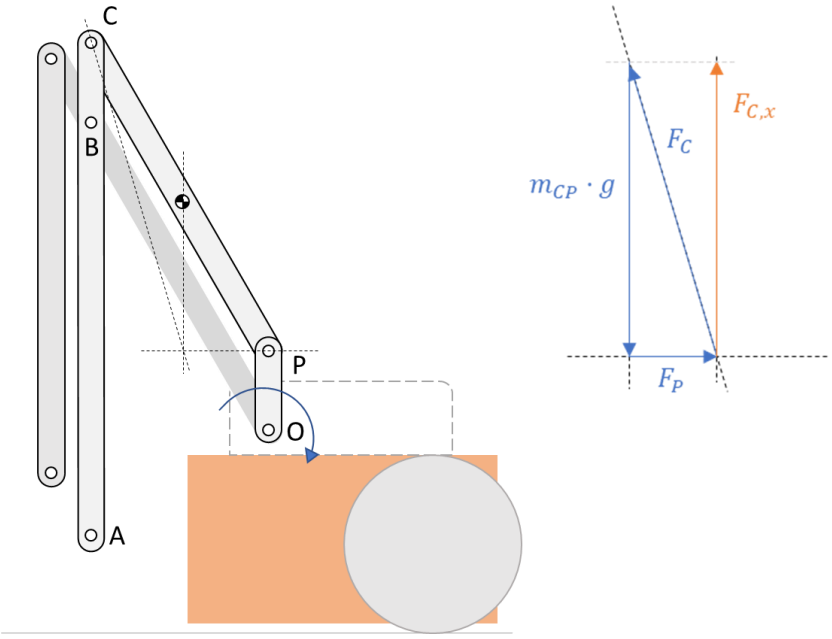


Figura 29. Estática gráfica sobre el antebrazo I.

En el caso en el que la pinza esta levantada, tampoco se sobrepasa el límite de los 90 grados mencionado anteriormente. En este caso, la barra OP no queda completamente perpendicular a la barra fija, y se puede apreciar en la figura como el peso de la barra favorece al movimiento haciendo que la proyección de la fuerza resultante de C sobre el plano perpendicular al segmento ABC sea mayor que la propia fuerza ejercida en el punto P por el motor.

En conclusión, el momento descrito en el caso ideal se considera válido para el movimiento favorable de la estructura.

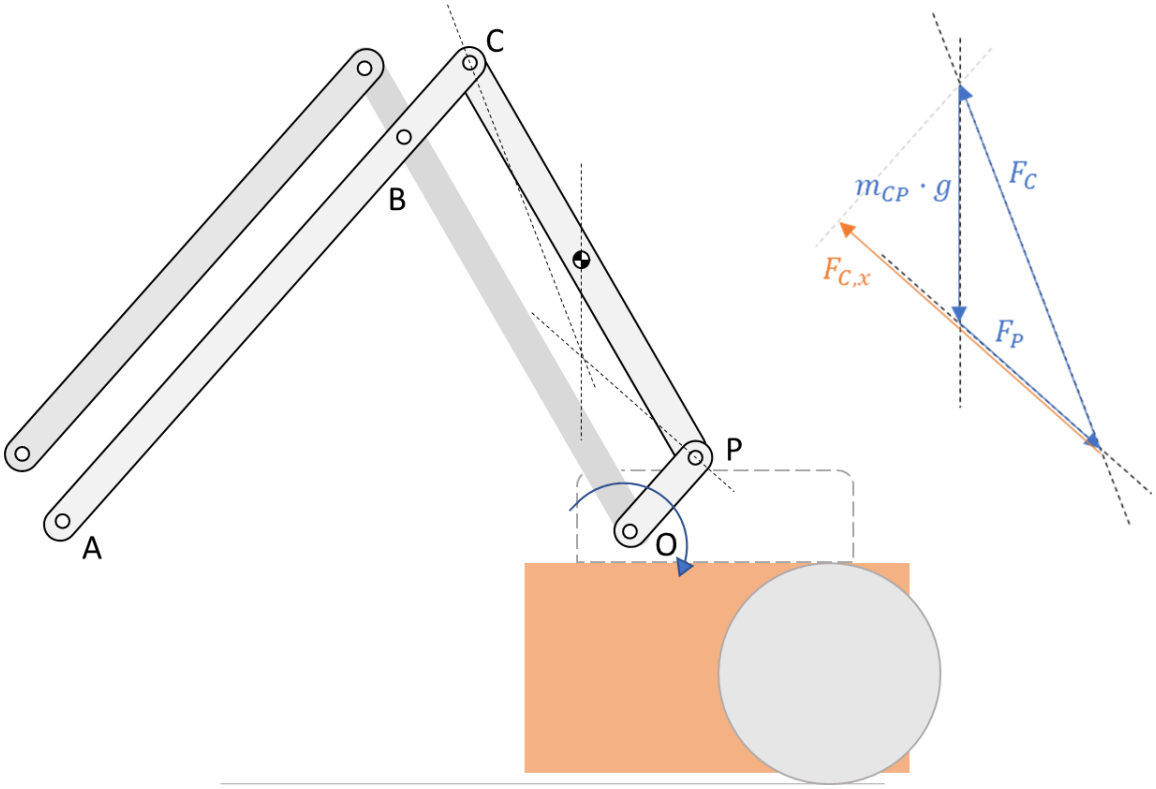


Figura 30. Estática gráfica sobre el antebrazo II.

5.2.2. Elección de componentes del brazo

Para aumentar la precisión del hombro, en este caso es necesario añadir un motor paso a paso. Los requisitos para este motor serán que el torque sea suficientemente grande como para mover la estructura y que sus características físicas (peso y tamaño) tengan valores pequeños. Las características eléctricas serán satisfechas por el *driver* que acompañara al conjunto para asegurar que el motor no supera la corriente máxima. El motor paso a paso será de tipo bipolar, para simplificar el cableado.

El *driver* elegido es el modelo *DRV8834*[34] compatible con motores bipolares paso a paso. Ofrece una intensidad máxima de 2 A y un rango de trabajo de tensión entre 2,5 y 10,8 V. En caso de que la intensidad supere 1,5 amperios, es necesario añadir un disipador de calor.

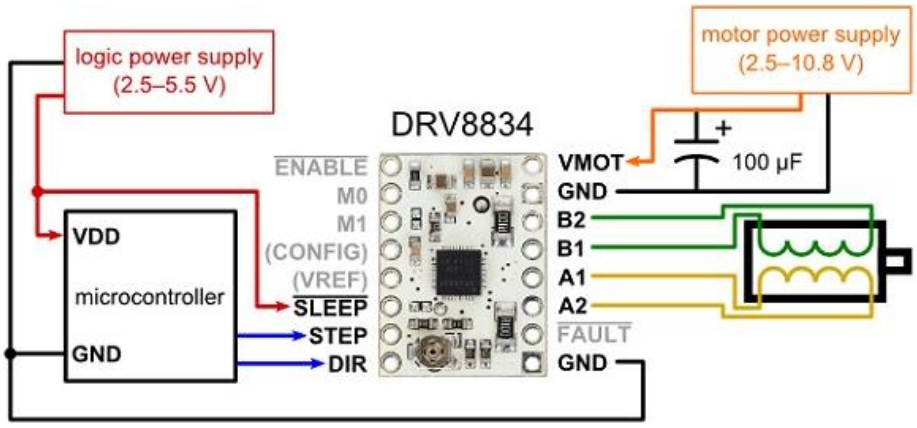


Figura 31. Conexión Driver DRV8834 con motor paso a paso.

La mayoría de las opciones encontradas compatible se han encontrado en el catálogo de motores paso a paso que ofrece *Kysan Electronics*[35], a excepción de la última opción encontrada en *Robotdigg*[36]. Los tamaños son iguales o inferiores a la regla NEMA 17.

Modelo	Voltaje (V)	Corriente (A)	R (Ω)	L (mH)	Torque (kg·cm)	Peso (g)	Tamaño (mm)	Precio (USD)
42BYGH10 (1124008)	12	0,4	30	58	3,8	240	39x24	17,50
42BYGH30 (1124010)	8,1	0,9	9	14	4,4	340	48x43	17,50
42BYGH3905 (1124023)	5,6	0,8	7	10	4,2	240	40x24	17,5
42BYGH48 (1124030)	4,2	1,5	2,8	4,8	5	340	48x17	17,5
57BYGH00 (1126009)	14,2	0,4	35,5	52	5,6	420	41x22	30,87
17HS6002	4,05	1,5	2,7	6,5	6,6	450	48x60	10

Tabla 4. Opciones valoradas para motor paso a paso.

La opción elegida es el motor 42BYGH3905 de *Kysan Electronics*. Se ha decidido así ya que el torque es suficiente para asegurar la estructura con un factor de seguridad de 1,39 y su corriente de 0,8 A no pondría en riesgo el conjunto por la posibilidad de altas temperaturas. Aun así, se recomienda optar por un sistema de disipación de calor en el *driver*, incluido en el paquete, para prevenir posibles riesgos.

La segunda mejor opción será también de la familia de *Kysan Electronics*, el motor paso a paso 42BYGH48. En este segundo caso, la corriente se eleva considerablemente, convirtiendo en obligatorio el uso de un disipador de calor. El torque es mayor, aunque también lo es su peso, pudiendo ocasionar dificultades en la movilidad del robot que se empleara como plataforma.

6. Software y programación

En este apartado se expondrá la programación de todas las partes para poder desarrollar las diferentes simulaciones. Se explicará también el código empleado.

Para el funcionamiento del proyecto, será necesario realizar dos programaciones diferentes. En primer lugar, se programará en Arduino IDE la placa ESP32-CAM, que contendrá las características necesarias para alterar la imagen y controlar el brazo robot a través de un servidor web que se creará a partir de un documento HTML. Por último, se programará con EdPy el robot Edison para su control en el plano horizontal.

El código completo para la placa ESP32 se encuentra al final de la memoria en el Anexo II.

6.1. Programación de la placa ESP32-CAM

El código usado para la placa está diseñado para ser cargado una sola vez a la placa. Una vez cargado, a placa será capaz de realizar todas las actividades planteadas. Puesto que la placa necesita estar conectada a una red Wifi, únicamente será necesario recargar el código al variar la dirección IP.

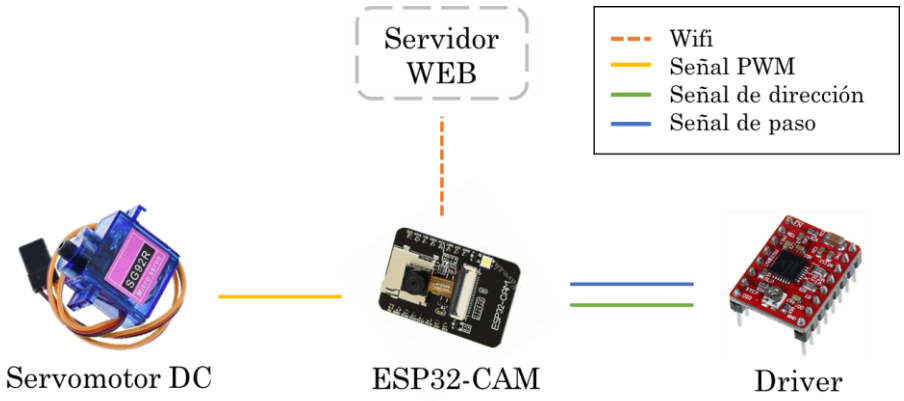


Figura 32. Esquema de programación en ESP32-CAM.

La placa estará conectada mediante tres pines al servomotor y driver ya mencionados. Estas salidas controlarán la señal PWM del servomotor en la pinza, la

señal de paso y la señal de dirección del driver, además de estar conectada al servidor web.

Como se ha mencionado anteriormente, la ESP32-CAM ofrece un código ejemplo en Arduino IDE que permite controlar la imagen desde un servidor web. Este código ha sido tomado como ejemplo y alterado, eliminando funcionalidades que no son necesarias y añadiendo propiedades nuevas para configurar el control del servomotor y del motor paso a paso.

6.1.1. Descripción del documento HTML

El servidor web se ha compuesto de un documento HTML y un código Arduino encargados de emitir y recibir las diferentes URI que identifican las peticiones de ambos elementos. En caso de que el usuario haga una petición desde un dispositivo móvil, esta será enviada a la placa, la cual procesará la petición y mandará una respuesta, que a su vez será recibida y procesada, y viceversa.

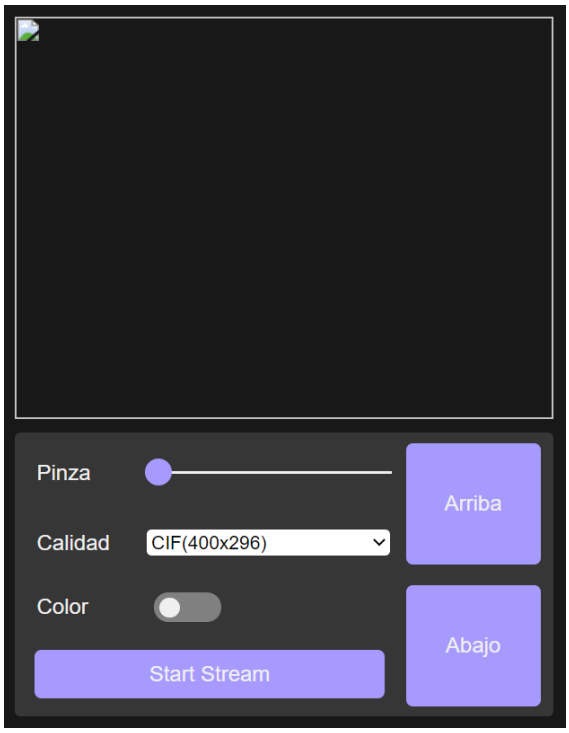


Figura 33. Diseño del control en el dispositivo.

El documento HTML contiene el diseño de la página web. En la figura anterior se muestra el diseño final, que permite la visualización de la imagen *streaming* en vivo, la edición de dicha imagen y el control del brazo robot conectado a la placa ESP32. El diseño se compone de un botón, dos pulsadores, una pestaña de selección, un pasador y un interruptor. La modificación de cada uno de estos elementos constituirá una petición o URI diferente compuesta por una variable *string* como elemento identificador y un valor, y la dirección de destino.

Función streaming

Una vez pulsado el botón nombrado como *Start Stream*, se mostrará la imagen en vivo y a su vez, se sustituirá el texto por *Stop Stream*. De esta manera, al volver a pulsar dicho elemento, la imagen desaparecerá, dejando un hueco negro en la pantalla. En este caso, no existe petición a la placa, ya que estas acciones están únicamente incluidas en el *script* del documento HTML. La imagen se recoge de la URI "*{document.location.origin + ':81'}/stream*". Esta función se ha heredado del código ejemplo.

Función sobre el control

Todos los controles se mandarán a través de la URI "*{document.location.origin}/control*". En este caso, también se mandará el identificador de la variable alterada y el valor final para esta variable. Luego, la función referida al control en la ESP32 clasificará la petición dependiendo del nombre de la variable y ejecutará las líneas de código correspondientes. A continuación, un extracto de código que muestra la petición del documento HTML.

```
const query = `${document.location.origin}/control?var=${el.id}&val=${value}`

fetch(query)
  .then(response => {
    console.log(`request to ${query} finished, status: ${response.status}`)
  })
```

En el extracto de código anterior, se puede apreciar como la variable *query* contiene tanto la dirección a la que está destinada como el valor e identificador del elemento

alterado. Se manda la petición con el comando *fetch* y se actualiza el estado en respuesta.

El pasador nombrado como *Pinza*, mandará una señal a la placa identificada como *servo*, junto con un valor dependiendo de la posición final del pasador. Esta petición será mandada a la placa, la cual responderá emitiendo una señal digital al pin de salida conectado a la señal PWM del servomotor. La posición inicial del pasador es 0 y su posición máxima tiene un valor de 255, porque se creará una señal PWM con una resolución de 8 bits.

```
<tr><td align="left">   Pinza </td>
    <td align="left" colspan="1">
    <input type="range" id="servo" min="0" max="255" value="0">
```

En el extracto anterior se puede apreciar cómo se especifica el tipo (*range*) y el nombre (*servo*) del elemento. En todos los casos se han seguido los mismos pasos. Esta descripción se hace al comienzo del cuerpo del HTML.

Los pulsadores nombrados como *Arriba* y *Abajo*, subirán o bajarán el brazo respectivamente al mantenerse pulsados. Una vez se dejan de pulsar, el brazo parará. Para esto, el HTML está diseñado para entender cuándo el pulsador está activo y cuándo deja de estarlo. En caso de que el pulsador "arriba" esté activado, manda un valor "1" bajo la variable "*stepper*". Si está activado el pulsador abajo, se manda un "2". Por último, en caso de que ninguno este activado, se mandará un "3". Dependiendo del valor recibido en la ESP32, se ejecutarán diferentes líneas de código.

Por último, las funciones del interruptor y de la pestaña de selección son las heredadas del código ejemplo. La primera, manda una señal a la placa que a partir de entonces mandará una imagen en escala de grises. La segunda, ofrece diferentes opciones para modificar la resolución de la imagen.

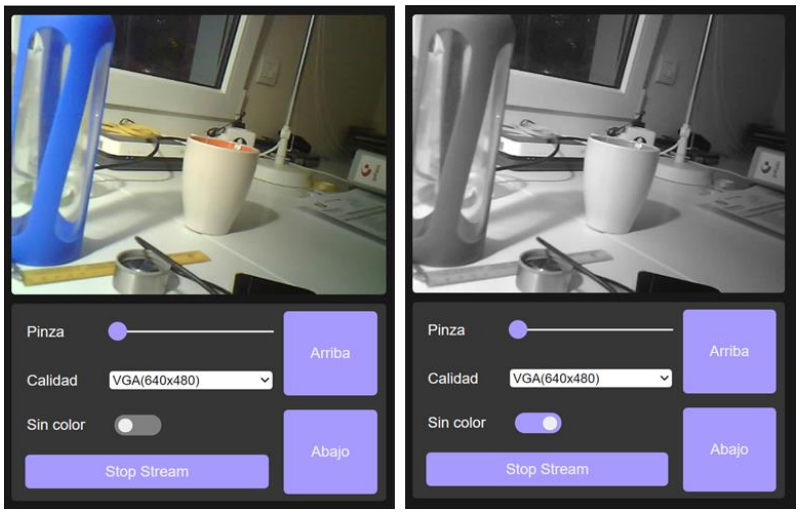


Figura 34. Visualización de la imagen alterada.

Función sobre los estados

Por último, también se comunica el estado inicial o actual de las variables. Esto se hace a partir del URI "*{document.location.origin}/status*". Solamente se consideran como estados el efecto especial sobre la imagen y la resolución.

6.1.2. Interpretación del código Arduino IDE

El proyecto en Arduino ha sido llamado "*ControlESP32CAM-ProyectoFinDeGrado.ino*" y llama a funciones que se encuentran en el archivo *.cpp* llamado "*FuncionesServidorWeb*". Ambos hacen uso de librerías ofrecidas por Arduino IDE específicas para servidores web en placas ESP32 y una librería propia del proyecto:

- La librería "*pins_y_HTML.h.*" contiene un listado que nombra a los diferentes pines para su posterior configuración. Además, contiene el documento HTML que constituye la web que servirá para el control de la placa. Este documento HTML ha sido traducido a Hexadecimal. El cambio a hexadecimal se ha realizado a partir de la web iCyberChef.com.
- La librería "*esp_camera.h*", al igual que el resto que van a ser nombradas a continuación, la ofrece Arduino IDE. Contiene las funciones que inicializan la cámara, configurándola a través de la interfaz I2C e inicializa la entrada I2S. además de las funciones que pueden leer los datos de la cámara.
- La librería "*Arduino.h*" permite la comunicación por el Serial Port, con el que se ha comprobado el funcionamiento del código al cargarse en la placa a través de los mensajes mandados con el comando *Serial.println*.
- "*Wifi.h*" es necesaria para poder conectar el elemento a la red inalámbrica, ya que se necesita el comando *Wifi.begin(ID, contraseña)* para ello.
- La última librería importante es "*esp_http_server.h*", la cual contiene la configuración por defecto del servidor HTTP. En ella también se muestra la estructura que define las URI y las peticiones y los comandos con los que se comunican al servidor HTTP.

En el archivo principal del proyecto, el código pide que se introduzca el nombre y la contraseña de la red wifi en la que se encuentra. Estas se introducen como variables. También, en este documento se configurara la cámara, se inicializaran los pines para el control de los motores y se llamará a la funciones *StartCameraServer()*, definida en el archivo "*FuncionesServidorWeb.cpp*". Una vez la cámara esté conectada al wifi y se hayan hecho las acciones correspondientes, se mandará al monitor serie un mensaje que contendrá la dirección IP del servidor web creado. Todo lo mencionado

anteriormente, está incluido en la función "*void setup ()*" del código, significando que únicamente se ejecutará una vez al subir el código.

La función *StartCameraServer()* está definida en el documento "*FuncionesServidorWeb.cpp*". Esta estructura y registra las URI de tipo HTTP_GET, ejecuta el servidor HTTP con la configuración por defecto mencionada anteriormente, y abre el hilo de comunicación. Se definen los puertos 80 y 81 para el servidor web y el servidor *streaming*, respectivamente. También se definen los diferentes *handlers* para cada URI.

- El nombrado "*index_handler*", manda el texto HTML como respuesta a la petición a partir del comando "*httpd_resp_send*".
- El handler "*status_handler*" responde con una declaración del estado actual de la resolución y el efecto sobre la imagen actual.
- El handler referido al control se llama "*cmd_handler*". Dependiendo del nombre de la variable que aparece en la URI, se mandarán señales digitales a los pines o se responderá al dispositivo alterando la imagen.
- Por último, el "*stream handler*" contiene un buble infinito para mantener un envío de imagen constante.

6.2. Programación del Robot Edison

La programación de este robot en *EdPy* es muy simple, aunque tiene de inconveniente que se debe cargar el código cada vez que se quiera cambiar de modalidad. Entre simulación y simulación, el robot deberá cargarse con un código diferente.

Por otro lado, el robot se puede programar a partir de la lectura de códigos de barras. Con el sensor que tiene en la parte inferior de la plataforma, lee estos códigos de barras y programa ciertas funciones predeterminadas. El control remoto es una de estas funciones, acelerando el proceso de programación en el caso de la simulación de daltonismo.

Figura 35. Código de barras ejemplo para la programación del Edison.

6.2.1. Simulación de la pérdida total de la vista

Para simular que el usuario ha perdido el sentido de la vista, el robot deberá tener la cámara desactivada, dejando la funcionalidad del robot reducida al control remoto. El usuario tendrá que dirigir el robot a ciegas y únicamente tendrá como guía las señales auditivas emitidas por el robot cuando este percibe un obstáculo con los sensores ultrasonido.

Es decir, para esta simulación únicamente será necesaria la programación del robot *Edison V2.0* para que siga las instrucciones del control remoto y mande una señal auditiva al detectar algún obstáculo.

El robot consta de un sistema de infrarrojos reflexivos. Esto quiere decir que está constantemente emitiendo luz infrarroja y que el sensor medirá los rayos que se reflejan

en las superficies. El sensor es capaz de definir si el obstáculo está a la derecha, izquierda o en frente. El sensor filtrará la señal reflejada y medirá la distancia a la que se encuentra el obstáculo. Es posible definir la distancia en la cual el sensor mande un aviso de obstáculo al sistema. Este sistema de detección de obstáculos funcionará mientras que los objetos tengan una altura igual o mayor al robot Edison, que son 3,5 centímetros aproximadamente.

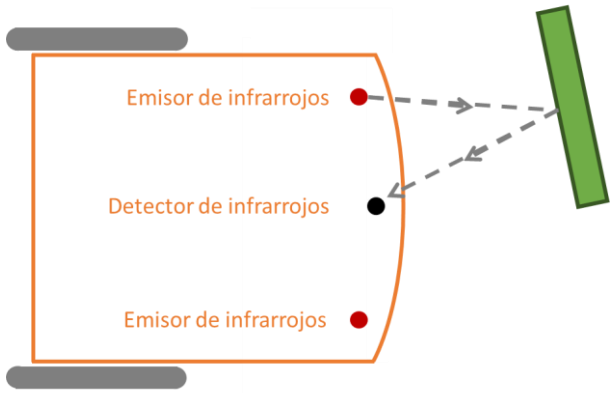


Figura 36. Detección de obstáculos por infrarrojos del robot Edison V2.0.

Para el código, simplemente se han relacionado las diferentes direcciones con el mando remoto enlazado y se ha especificado la respuesta que debe dar el robot al encontrarse cualquier objeto. Aunque sea posible, no se ha especificado una respuesta dependiendo de la ubicación del obstáculo, sino que se dará la misma señal acústica para cualquier ubicación del obstáculo para darle más dificultad a la prueba. A continuación, se muestra el código empleado para la programación del robot.

```
import Ed
Ed.EdisonVersion = Ed.V2
Ed.DistanceUnits = Ed.CM
Ed.Tempo = Ed.TEMPO_MEDIUM
Ed.ObstacleDetectionBeam(Ed.ON) #Encender deteccion de obstaculos
obstacle = 0

while True: #bucle infinito
    if Ed.ReadRemote() == Ed.REMOTE_CODE_NONE:
        pass
    elif Ed.ReadRemote() == Ed.REMOTE_CODE_1:    # HACIA DELANTE
        Ed.Drive(Ed.FORWARD, 5, Ed.DISTANCE_UNLIMITED)
    elif Ed.ReadRemote() == Ed.REMOTE_CODE_2:    # HACIA ATRAS
        Ed.Drive(Ed.BACKWARD, 5, Ed.DISTANCE_UNLIMITED)
    elif Ed.ReadRemote() == Ed.REMOTE_CODE_3:    # HACIA LA DERECHA
        Ed.Drive(Ed.FORWARD_RIGHT, 5, Ed.DISTANCE_UNLIMITED)
    elif Ed.ReadRemote() == Ed.REMOTE_CODE_4:    # HACIA LA IZQUIERDA
        Ed.Drive(Ed.FORWARD_LEFT, 5, Ed.DISTANCE_UNLIMITED)
    else:    # PARAR
        Ed.Drive(Ed.STOP, 1, 1)

    #actualizamos señal de obstaculo
      obstacle = Ed.ReadObstacleDetection()
      if obstacle>Ed.OBSTACLE_NONE: #en caso de que haya obstaculo
            #beep and back up
            Ed.Drive(Ed.BACKWARD, Ed.SPEED_5, 3)
            Ed.PlayBeep()
            Ed.ReadMusicEnd()
```

6.2.2. Simulación de daltonismo

Para simular la alteración visual de los colores, esta vez se necesita vincular la cámara al conjunto. El usuario únicamente tendrá el control remoto y le llegará la imagen con los colores distorsionados de la cámara.

Para esto, en el dispositivo móvil se activará el efecto de imagen de escala de grises mientras que se mantiene activado únicamente el control remoto. Por tanto, en este caso, se simplifica el código empleado para el caso anterior, eliminando la detección de obstáculos.

6.2.3. Simulación de movilidad reducida en las extremidades

Para simular el caso de una movilidad reducida en las extremidades inferiores, se debe alterar el código EdPy, reduciendo la velocidad del motor. El robot con el código alterado simulará así una desventaja en cuanto a velocidad.

En el caso de la simulación de la movilidad reducida del brazo, el control de brazo por parte del código quedará inhabilitado. El usuario no podrá usar las pinzas o el motor ligado al antebrazo.

7. Consumo eléctrico

En este capítulo se expondrá un estudio eléctrico del sistema. Se ha decidido dividir por diferentes partes dependiendo del voltaje necesario. Existen unas baterías independientes para el robot Edison, el conjunto ESP32-CAM y el controlador que servirá al motor paso a paso.

A todas las pilas se les puede unir en serie para aumentar su voltaje y unirlas en los extremos a un conector SM2P. A este conector se le puede enchufar un cable con terminación USB. De esta manera no será necesario sacar las pilas recargables del robot para que sean recargadas, sino que se podrá conectar un cargador.

Como se expone en los siguientes apartados, todas las propuestas se componen de pilas recargables NiMH. Al no ser desechables no son tan perjudiciales para el medio ambiente como pueden ser las pilas alcalinas de un único uso. Además, los dos tipos de baterías recargables más conocidos son las pilas níquel-cadmio (NiCd) y pilas de níquel-hidruro metálico (NiMH). El cadmio es un material caro y dañino para el medio ambiente, por tanto, las pilas de hidruro metálico se alinean mejor con los objetivos del proyecto. Además, las pilas de hidruro metálico tienen una capacidad de carga mucho mayor que las baterías de cadmio y se ven menos afectados por el efecto memoria.

7.1. Alimentación del robot Edison

Como se ha comentado anteriormente, el robot que se empleará como plataforma se alimenta a base de 4 pilas AAA. El robot permite la posibilidad de utilizar pilas alcalinas de único uso, o utilizar pilas recargables. Sin embargo, únicamente permite las pilas recargables de material níquel-metal hidruro. Esta última limitación no supone ningún inconveniente ya que este tipo es menos dañino al medio ambiente y tienen mejor rendimiento.

Debido al objetivo ecológico del proyecto se decidirá alimentar el robot a partir de las pilas recargables de níquel-hidruro metálico. Además, supondrá un beneficio económico a largo plazo al poder ser recargado y reutilizado en vez de sustituido.

7.2. Alimentación de la placa ESP32-CAM

La placa ESP32 estará conectada al sensor de imagen, el servomotor y el controlador del motor paso a paso. Consume 30 miliamperios (mA) mientras que la conexión bluetooth o wifi este desactivada y 180 mA mientras que este activada. Según la ficha técnica de la placa, el máximo de corriente de salida para pines digitales es de 40 miliamperios.

En el caso de la cámara, esta debe estar alimentada por la cámara a 3,3 voltios. La corriente máxima consumida por el sensor de imagen es de 40 mA mientras que la típica es de 30 miliamperios.

El *Micro Servo Digital 9g SG92R* de Tower Pro[37] debe estar alimentado a 5 voltios y consumirá 10mA en reposo y entre 100mA y 300mA en plena actividad. Puede llegar a consumir 650 mA al ejercer grandes esfuerzos. Al estar alimentado directamente de la placa, aun estando en reposo la placa deberá suministrar corriente.

El controlador *DRV8834* estará conectado a la placa a partir de dos pines digitales que definirán la dirección y avance del motor paso a paso. El pin de dirección estará activado para un sentido y desactivado para el contrario. En cambio, el pin de avance estará activo cada vez que el motor este en movimiento. Según las características del controlador *DRV8834*, la corriente de entrada a 3,3 voltios es de 33 microamperios, mientras que a 0 voltios es de 1 microamperio. Al estar conectado a una fuente de alimentación diferente, en caso de que se enfrente a un gran esfuerzo, este suplemento de energía no se verá reflejado en la placa, ya que lo sustraerá de la batería auxiliar.

En la siguiente tabla se recogen los diferentes consumos de energía de la batería de la placa en diferentes modos de operación. En el caso del controlador, se ha querido considerar un escenario en el que este constantemente activo. Se ha decidido mantener el movimiento constante, por tanto, el pin de paso estará activado en todo momento, recibiendo 33 mA en todo momento de la placa. Por el contrario, el pin de dirección variará, y estará activado la mitad de cada ciclo, ya que necesita la misma cantidad de tiempo para moverse en una dirección que en otra. En total, los pines dirigidos al controlador envían 49,5 mA en todo momento.

Modo de operación	ESP32	Cámara	Servomotor	Controlador	Consumo (mA)
Brazo en reposo	Activo	Activo	Resposo	Reposo	221
Funcionamiento normal	Activo	Activo	Activo	Activo	459,5
Funcionamiento crítico	Activo	Activo	Gran esfuerzo	Activo	909,5

Tabla 5. Consumo eléctrico de la ESP32 en diferentes modos de operación.

En conclusión, suponiendo que la conexión *streaming* (cámara y wifi) esta activada en todo momento, la corriente suministrada es de 220mA. Suponiendo que el brazo robot no está en reposo el consumo aumenta a 459,5mA. Por último, considerando el caso más crítico en el que el motor está activo permanentemente (pin de paso activo) y varía de dirección cada vez que llega a un tope (pin de dirección activo mitad de ciclo) y el servomotor está ejerciendo un gran esfuerzo, la corriente se eleva a 909,5mA.

Las baterías recargables Ni-MH AA de *EBL* contienen 1,2 voltios cada una con una capacidad de 2800mAh. Esto significaría que el robot podría funcionar durante más de tres horas en su estado más crítico o durante más de 6 horas en su funcionamiento normal. También duraría más casi 13 horas si no se usase el brazo.

7.3. Alimentación del Driver

El modelo de motor paso a paso *Kysan Electronics* 42BYGH3905 que finalmente se ha seleccionado para el brazo del robot, debe estar alimentado por 5,6 voltios y 0,8 amperios para su funcionamiento.

Para controlar que no se excede la intensidad de corriente límite en el motor se decidió emplear el controlador DRV8834[38]. Este debe estar alimentado con una batería cuyo voltaje este comprendido entre los 2,6 y 10,8 voltios para funcionar. Sin embargo, como el motor debe funcionar a 5,6 voltios, el controlador debe estar alimentado como mínimo a 5,6 voltios. El controlador será el encargado de mandar una corriente de 0,8 amperios por cada fase del motor. Al ser bipolar, tiene dos fases y por tanto la corriente total será de 1,6 amperios. Además, a esta corriente hay que sumarle la corriente de

abastecimiento para la operación del controlador, que llega a los 4 miliamperios en condiciones normales y 2 microamperios en estado de reposo.

En conclusión, el controlador del motor debe estar alimentado por una batería de entre 5,6 y 10,8 voltios que pueda suministrar 1,604 amperios. Con las pilas recargables mencionadas en el apartado anterior, el brazo tendría dos horas de carga. Se deberían unir al menos 5 pilas de 1,2 voltios para llegar al voltaje deseado.

8. Pruebas

En este capítulo se explicarán brevemente las pruebas realizadas y las pruebas por realizar para el funcionamiento del proyecto

8.1. Comprobación del funcionamiento de la cámara

Para realizar una comprobación de la funcionalidad de la cámara, se puede compilar un código predeterminado ofrecido por Arduino IDE. Para ello, únicamente es necesario instalar o actualizar las tarjetas ESP32 en Arduino IDE y compilar el código '*CameraWebServer*'. El código final del proyecto está basado en dicho código ya que necesita la conexión al servidor web.

Antes de compilar es necesario modificarlo introduciendo el nombre de la señal Wifi y la contraseña y descomentar el módulo de cámara *Ai-Thinker*.

Para subir el código a la tarjeta es necesario conectarla al ordenador haciendo uso del convertidor mencionado anteriormente. En el siguiente esquema se muestra la conexión de ambos dispositivos.

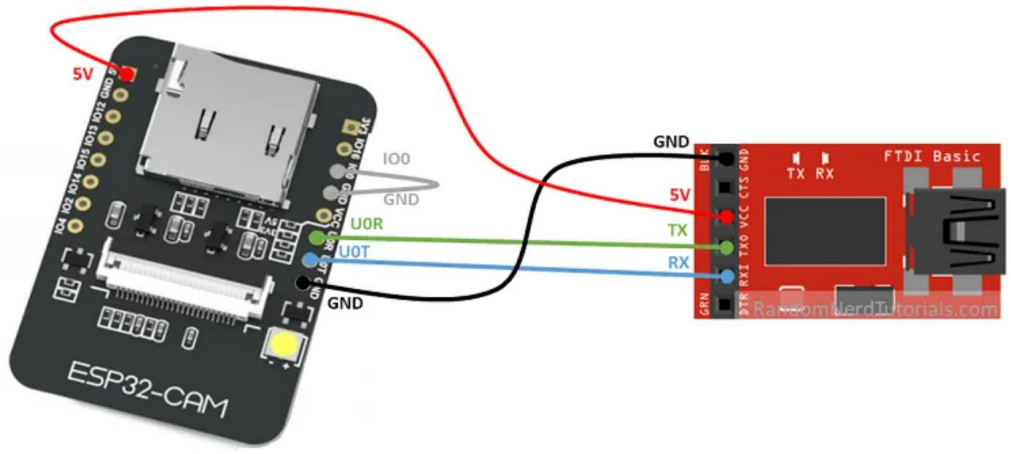


Figura 37. Mapa de conexión de ESP32-CAM y convertidor USB-Serial TTL.

Como se puede apreciar en la imagen anterior, la cámara tiene que estar alimentada por una batería en todo momento. El pin UOT de la cámara debe estar conectado al pin RX del adaptador y el UOR de la cámara al TX del adaptador. Durante la carga de datos, el pin IO0 tiene que estar puenteado a GND.

Antes de subir el código a la placa, se debe pulsar el botón RESET de la parte posterior de la placa. Una vez el código se haya subido, se debe quitar el puente entre GND y IO0 y volver a presionar el botón RESET.

Al realizar este último paso, en el monitor serie del ordenador debe aparecer la dirección IP con que se accederá posteriormente al servidor.

8.2. Simulación del movimiento del mecanismo

Para comprobar el funcionamiento de las diferentes partes se ha optado por simular el conjunto en el programa *Solid Edge* versión 2019. Este programa simula el movimiento que realizaría el mecanismo al introducir motores rotativos en los ejes especificados en apartados anteriores. Se ha simulado con la opción "movimiento físico", que permite una aproximación más cercana a la realidad. Se ha comprobado que no existen colisiones que puedan afectar al movimiento.

Conectando un motor rotativo en uno de los engranajes de la pinza, se ha podido comprobar que la pinza se abre y se cierra dependiendo de la dirección de giro del motor. La pinza se ha comportado favorablemente, abriendo y cerrando sin sufrir colisiones. También se ha comprobado el equilibrio de la estructura en situación de reposo frente a la fuerza de la gravedad. La pinza se mantiene en perfecto equilibro sin subir inclinaciones en ningún eje.

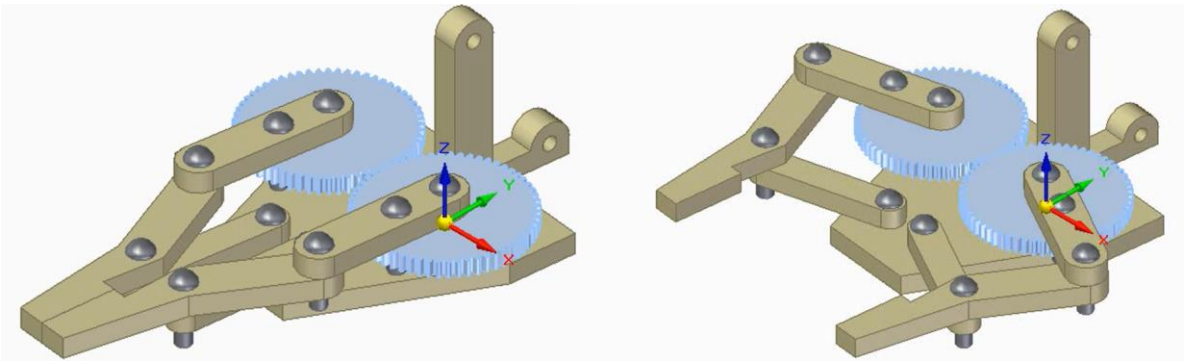


Figura 38. Simulación cinemática de la pinza.

Por último, se ha comprobado también el funcionamiento del antebrazo haciendo una simulación parecida al caso anterior. En este caso, el motor se ha situado en el eje que une el brazo con la base del robot plataforma. Dependiendo de la dirección del motor, el brazo sube o baja. El desplazamiento angular máximo permitido según un estudio cinemático es de 90 grados, aunque según el estudio estático teórico este ángulo se debe limitar a 60 grados.

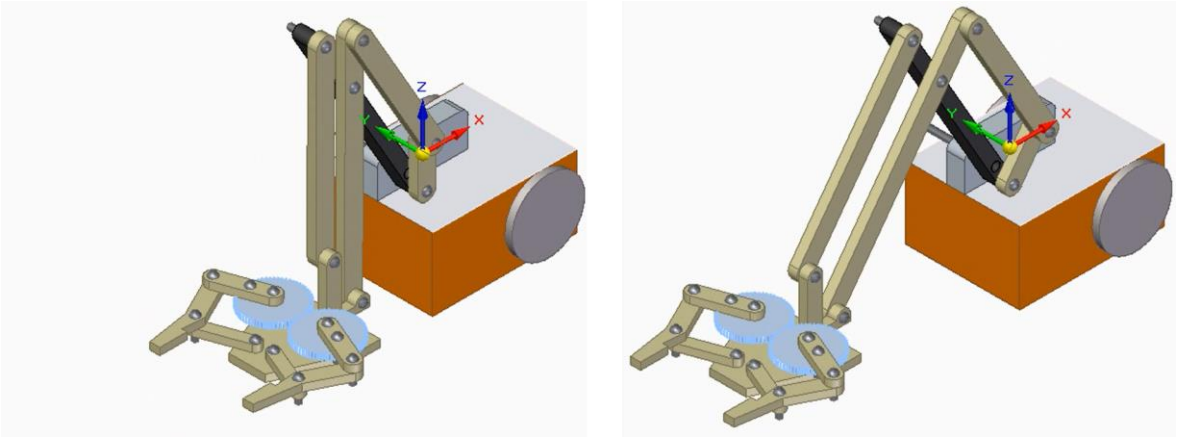


Figura 39. Simulación cinemática del brazo.

9. Presupuesto

En este capítulo se llevará a cabo una estimación tanto económica como de tiempo empleado en el trabajo. Se incluirá el coste material que conlleva el proyecto y las horas que se deben dedicar a realizarlo

9.1. Coste de componentes del Hardware

En este apartado se expondrá el coste de la parte Hardware, incluyendo tanto el precio de los componentes como el de desarrollo.

9.1.1. Estimación de coste del Robot Edison V2.0.

Se incluyen como coste del Robot Edison V2.0, todo lo esencial para su funcionamiento de manera independiente al resto de la estructura. Es decir, se incluye en este apartado el coste del aparato en sí, el mando de infrarrojos que se emplea para su control y el coste de las baterías para su alimentación.

Los precios de los objetos mencionados según la compañía Amazon son los siguientes. En total, el coste estimado del conjunto es de 52,73 euros.

Producto	Precio (€)
Edison V2.0. Stem Robot	44,99
Mando universal de infrarrojos mini	3,29
4 pilas AAA recargables Ni-MH Green Cell	4,45
Total	52,73

Tabla 6. Estimación de coste de Robot Edison V2.0.

9.1.2. Estimación de coste de ESP32-CAM y cámara

En este apartado contiene la estimación del grupo ESP32-CAM, que contiene el coste de la placa ESP32 junto con el sensor de imagen OV2640, y el convertidor serial

TTL para su conexión por USB a un ordenador. No es posible comprar la ESP32-CAM por separado, sino que se debe comprar en conjunto con el sensor de imagen OV2640.

Los precios según la compañía Amazon son los siguientes. En total, el conjunto tiene un valor de 25,24 euros.

Producto	Precio (€)
ESP32-CAM + OV2640	8,99
Convertidor Serial TTL USB	6,69
Pack de 40 cables de puente	1,99
4 pilas AA recargables Ni-MH Duracell 1300mAh	7,57
Total	25,24

Tabla 7. Estimación de coste de conjunto ESP32-CAM.

9.1.3. Estimación de coste de brazo mecánico

En este apartado se incluye la estimación de coste del conjunto electrónico y mecánica del brazo robot.

En este caso, el precio estimado el motor paso a paso se ha encontrado en la página web de *Kysan Electronics* y el precio del micro servo SG92R en la web banggood.com. El precio del controlador DR8834 se ha buscado en la web del *Pololu*, su fabricante. Las pilas se han encontrado en la web de la compañía Amazon.com. La estimación total es de 43,18 euros.

Producto	Precio (€)
Micro servomotor digital SG92R	4,51
Motor paso a paso Kysan Electronics 42BYGH3905	15,44
Driver DR8834 Low Voltage	5,24
Pilas recargables AA 6V 2400mAh con conector USB SM2P	17,99
Total	43,18

Tabla 8. Estimación de costes de la parte electrónica del brazo robot.

Por otro lado, se debe incluir el coste económico de la parte mecánica del brazo, incluyendo materiales. En la siguiente tabla se incluirán el coste aproximado de los 10 pasadores planteados para la pinza, más los 8 pasadores necesarios para la construcción del brazo robot. También se debe incluir en valor de los engranajes que se quiera emplear y el coste del material haciendo una aproximación de la superficie final del mecanismo. La superficie del mecanismo del brazo se estima de menos de 8700 milímetros cuadrados, de los cuales casi 3890 corresponden a la superficie de la pinza.

Los datos relacionados con el coste de los pasadores de 3 mm de diámetro se han encontrado en Amazon.com, mientras que los pasadores restantes de diámetro 4 milímetros son ofrecidos por Leroy Merlin. Los engranajes se han buscado en la página web de RS PRO, al igual que las planchas de aluminio de 2 milímetros de espesor. La información sobre el cartón gris se ha encontrado en papelesespeciales.es.

Producto	Precio (€)
10 pasadores acero 3 mm de diámetro, 10 mm de longitud	4,83
10 pasadores acero 4 mm de diametro, 50 mm de longitud	1,39
2 engranajes rectos M0,5 - 60 dientes	9,24
Engranaje recto M0,5 - 25 dientes	2,84
Dos planchas de carton gris 70x105cm	9,1
Lámina aluminio 300x600x2 mm	11,18
Total	38,58

Tabla 9. Estimación de costes de la parte mecánica del brazo.

9.1.4. Estimación de perfiles profesionales

A continuación, se recogen las actividades relacionadas al hardware del proyecto.

- Elección de componentes y diseño del proyecto en cuestión.
- Diseño del brazo robot, el cual incluye una simulación de su comportamiento.
- Montaje del brazo robot.
- Montaje de todos los sistemas.

Se considera que estas actividades deben ser repartidas entre profesionales en el ámbito de la robótica y electrónica. El salario medio de un técnico de robótica es de 15,8 euros a la hora. El salario medio de un ingeniero electrónico es de 29,69 euros al año.

Tarea	Responsable	Horas	Coste hora (€)	Coste total (€)
Diseño del proyecto	Ingeniero electrónico	40	29,69	1187,6
Diseño del brazo robot	Ingeniero electrónico	70	29,69	2078,3
Montaje del brazo robot	Técnico en robótica	20	15,8	316
Montaje de todos los sistemas	Técnico en robótica	10	15,8	158
Total		140		3739,9

Tabla 10. Estimación de perfiles profesionales Hardware.

9.1.5. Estimación total Hardware

A continuación, se muestra el sumatorio total de los costes de perfiles profesionales con los costes referidos al material.

Estimaciones	Coste (€)
Estimación de coste de Perfiles Profesionales	3739,9
Estimación de coste del Robot Edison V2.0. + Equipamiento	52,73
Estimación de coste de componentes de brazo robot + materiales	38,58
Estimación de coste de ESP32-CAM + Equipamiento	25,24
Total	3856,45

Tabla 11. Estimación de coste total Hardware.

9.2. Coste de componentes Software

Es necesario realizar la misma estimación para los componentes software del proyecto. Al ser los entornos empleados para la programación gratuitos, únicamente es necesario estudiar el coste de los perfiles profesionales. Las tareas que se deben llevar a cabo son las siguientes:

- Diseño de servidor web útil para controlar el brazo robot.
- Diseño del control de las diferentes partes del brazo robot en Arduino IDE.
- Diseño del control del robot Edison en EdPy.
- Pruebas.

En este caso, únicamente se ve necesario añadir un perfil de programador que este familiarizado con la implementación de un servidor web. El salario estimado de un programador es de 16,60 euros por hora.

A continuación, se muestra una estimación de las horas invertidas en el proyecto junto con el coste total estimado.

Tarea	Responsable	Horas	Coste hora (€)	Coste total (€)
Diseño de servidor web	Programador	115	16,60	1909
Diseño control del brazo	Programador	50	16,60	830
Diseño control robot Edison	Programador	20	16,60	332
Pruebas	Programador	25	16,60	415
Total		210		3486

Tabla 12. Estimación de coste total Software.

9.3. Coste total del proyecto

Al haber realizado las estimaciones referidas al Hardware y Software del proyecto, se puede hacer una estimación completa de tiempo y coste económico.

Tarea	Horas	Coste total (€)
Estimación de coste Hardware	140	3856,45
Estimación de coste Software	210	3486
Total	350	7342,45

Tabla 13. Estimación de coste total.

10. Conclusiones y Trabajos futuros

En este capítulo se exponen las conclusiones del proyecto (Sección 10.1.) junto con las aplicaciones que no se han podido desarrollar (Sección 10.2.).

10.1. Conclusiones

Una vez terminado el proyecto, se puede decir que se ha cumplido con el objeto principal de diseñar un robot que pueda simular ciertas discapacidades. Las discapacidades que finalmente se han podido simular son la ceguera, el daltonismo y la reducción de las propiedades motoras de extremidades inferiores y superiores.

Uno de los objetivos del proyecto era realizar un robot que tuviese el mínimo impacto negativo en el medio ambiente. Una vez finalizado el proyecto se puede admitir que este objetivo, no se ha cumplido totalmente debido a la existencia de pastico en el robot Edison empleado en la plataforma. Sin embargo, este robot se puede sustituir por un robot de diseño similar, pero con mejores características ecológicas, que contenga a su vez un sensor ultrasonido o de infrarrojos que pueda detectar obstáculos y por tanto sirva de ayuda a la navegación.

Por otro lado, los materiales empleados para el brazo robot sí que se han cumplido, ya que las fuerzas que el robot debe resistir son pequeñas en comparación a las características del aluminio y del cartón gris contracolado.

Finalmente, debido al cumplimiento del objetivo principal y la intención de diseñar un robot ecológico, se han cumplido dos propuestas de los Objetivos de Desarrollo Sostenible de Naciones Unidas.

10.2. Trabajos futuros

En este apartado se expondrán ideas sobre cómo se podría expandir este proyecto para ocasiones futuras.

10.2.1. Desarrollo de una interfaz que globalice el control

Una de las propiedades que han faltado en este proyecto es la posibilidad de agrupar todos los controles en uno solo. En próximos desarrollos de este proyecto, debería plantearse un control que pueda adoptar el papel de controlar el robot plataforma y el brazo robot desde un mismo dispositivo.

Actualmente, el robot que se ha utilizado como plataforma es el robot Edison V2.0. que permite el control remoto por infrarrojos. Por otro lado, la placa ESP32-CAM y todas sus conexiones se controlan a partir de un servidor web.

La ventaja del proyecto es que el robot Edison es completamente independiente a la placa ESP32-CAM, pudiendo ser sustituido por otro robot que sirva de plataforma. Se podría elegir un robot capaz de comunicarse con un servidor web, para que todo el control del conjunto se pueda realizar desde cualquier dispositivo con conexión a internet.

10.2.2. Desarrollo de otras simulaciones

Para el desarrollo del robot y conseguir que se asemeje más a la realidad, se podrían mejorar las simulaciones ya realizadas o crear unas nuevas.

En el caso de la simulación del daltonismo, en este proyecto se ha optado por un efecto de escala de grises para anular por completo los colores de la imagen. En un futuro, se podría desarrollar esta deformación de imagen para que los colores del robot queden anulados permitiendo una simulación mucho más parecida al daltonismo.

Para aumentar el número de simulaciones en el robot, se podría intentar realizar una simulación de una discapacidad auditiva, en el que los robots tengan introducido un micrófono y puedan mandar la señal de audio al dispositivo de control para que el usuario pueda escucharlo. El audio se vería afectado dependiendo en caso de simular una discapacidad auditiva.

11. Referencias

[1] Instituto Nacional de Estadística (2008). *Panorámica de la discapacidad en España. Encuesta de Discapacidad, Autonomía personal y situaciones de Dependencia*. Recuperado de: https://www.ine.es/ss/Satellite?L=&c=INECifrasINE_C&cid=1259924962561&p=1254735116567&pagename=ProductosYServicios%2FPYSLayout. [fecha de extracción: 10/01/2020]

[2] Fundación Llorente y cuenca. (2017, diciembre 3). *"La discapacidad no significa incapacidad"*. Recuperado de: https://www.fundacionllorenteycuenca.org/2017/12/03/la-discapacidad-no-significa-incapacidad/. [fecha de extracción: 11/01/2020]

[3] *Todo sobre STEAM: qué es, cómo funciona y para qué sirve.* (2019, 4 septiembre). Lúdilo. Recuperado de: https://www.ludilo.es/blog/todo-sobre-steam-que-es-como-funciona-y-para-que-sirve/ [fecha de extracción: 03/05/2020]

[4] PlasticsEurope. (s. f.). *Plásticos – Situación en 2019*. Recuperado de: https://www.plasticseurope.org/es/newsroom/neuigkeiten/plasticos-situacion-en-2019 [fecha de extracción: 04/05/2020]

[5] Greenpeace. (s. f.). *Datos sobre la producción de plásticos*. Greenpeace España. Recuperado de: https://es.greenpeace.org/es/trabajamos-en/consumismo/plasticos/datos-sobre-la-produccion-de-plasticos/ [fecha de extracción: 04/05/2020]

[6] Designthemes. (2018, julio 10). *Los protocolos TX RX de Radio control*. Tienda y Tutoriales Arduino. Recuperado de: https://www.prometec.net/protocolos-tx-rx-de-radio-control/ [fecha de extracción: 10/02/2020]

[7] *Guía Básica Protocolos de Comunicación R/C: PWM, PPM, SBUS, DSM2, DSMX…* (2017, julio 28). FpvMax. Recuperado de: http://fpvmax.com/2017/07/28/protocolos-comunicacion-drones/. [fecha de extracción: 10/02/2020]

[8] Ecuador. (2012, octubre 17). *Comunicaciones infrarrojas*. SlideShare. Recuperado de: https://www.slideshare.net/lilys9018/comunicaciones-infrarrojas-14761006 [fecha de extracción: 10/02/2020]

[9] *Inputs, outputs and sensors - get to know your Edison robot*. (s. f.). Meet Edison. Recuperado de: https://meetedison.com/edison-robots-sensors/ [fecha de extracción: 01/02/2020]

[10] Makeblock. (s. f.). *Robot Kits for Kids: mBot | Makeblock - Global STEAM Education Solution Provider*. Recuperado de: https://www.makeblock.com/mbot [fecha de extracción: 01/02/2020]

[11] Villanueva, A. (2018, marzo 17). *mCore: mapeo de puertos y programación*. Mecatrónica Lab. Recuperado de: http://www.mecatronicalab.es/mcore-mapeo-de-puertos-y-programacion/ [fecha de extracción: 02/02/2020]

[12] *What is web socket and how it is different from the HTTP?* (2019, diciembre 4). GeeksforGeeks. Recuperado de: https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/ [fecha de extracción: 01/06/2020]

[13] *Buy a Camera Module V2 – Raspberry Pi*. (s. f.). Raspberry Pi. Recuperado de: https://www.raspberrypi.org/products/camera-module-v2/ [fecha de extracción: 05/02/2020]

[14] Ai-Thinker (s. f.). *ESP32-CAM Wi-Fi + BT SoC Module V1.0* [documento PDF]. Recuperado de: https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf [fecha de extracción: 05/02/2020]

[15] Prasad, K. (s. f.). *OV2640 Camera Module: Documentation for Tang Primer*. SiPEED. Recuperado de: https://tang.sipeed.com/en/hardware-overview/ov2640-camera/ [fecha de extracción: 05/02/2020]

[16] *YI Technology*. (s. f.). YI Technology. Recuperado de: https://www.yitechnology.com/yi-home-20 [fecha de extracción: 05/02/2020]

[17] *Xiaomi Xiaofang*. (s. f.). Xiaomi en casa. Recuperado de: https://xiaomiencasa.com/gadgets/domotica/xiaomi-xiaofang.html [fecha de extracción: 05/02/2020]

[18] V380 Camera. (s. f.). *V380 wifi mini night vision hidden cctv spy voice intercom wifi camera*. Recuperado de: https://v380-camera.com/products/v380-360eyes-mini-hidden-cctv-spy-wifi-camera [fecha de extracción: 05/02/2020]

[19] ZOOBOTICS. (s. f.). *ZURI 01 Paperbot System*. Recuperado de: http://zoobotics.de/project/zuri-01-3/ [fecha de extracción: 05/04/2020]

[20] Gray, B. (2019, junio 20). *Introducing the MeArm*. MeArm - Pocket Sized DIY Robotic Arm Kits. Recuperado de: https://mearm.com/2019/06/14/welcome-to-my-blog/ [fecha de extracción: 05/04/2020]

[21] Robert, H. (s. f.). *Is Fiberboard Recycleble?* SFGATE. Recuperado de: https://homeguides.sfgate.com/fiberboard-recyclable-79237.html [fecha de extracción: 05/05/2020]

[22] Leicester Press Office (2013, octubre 31). *New recyclable MDF could help solve UK waste problem*. Recuperado de:https://www2.le.ac.uk/offices/press/press-releases/2013/october/new-recyclable-mdf-could-help-solve-uk-waste-problem?utm_medium=website&utm_source=archdaily.com [fecha de extracción: 05/05/2020]

[23] *ESP32-CAM Video Streaming and Face Recognition with Arduino IDE*. (2020, abril 30). Random Nerd Tutorials. Recuperado de: https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/ [fecha de extracción: 31/05/2020]

[24] Omnivision (2006, febrero 28). *Advanced information preliminary datasheet. OV2640 color CMOS UXGA (2.0 MegaPixel) CameraChip™ with OmniPixel2™ Technology* [documento PDF]. Recuperado de: https://www.uctronics.com/download/cam_module/OV2640DS.pdf [fecha de extracción: 31/05/2020]

[25] *Crear Servidor Web con Arduino*. (2014, January 24). DIYMakers. Recuperado de: http://diymakers.es/crear-servidor-web-con-arduino/ [fecha de extracción: 31/05/2020]

[26] *Cardboard paper manufacturers & duplex paper board manufacturers*. (s. f.). New Bamboo Paper. Recuperado de: https://www.greypaperboard.com/applications [fecha de extracción: 05/05/2020]

[27] RDM (2020 enero). *Ovaro 688 grey board GK* [documento PDF]. Recuperado de: http://rdmgroup.com/wp-content/uploads/2018/04/OVARO-688-4.pdf [fecha de extracción: 05/05/2020]

[28] *MCS Multicolor Spezial* (s. f.). MM Karton. Recuperado de: https://digi.mm-karton.com/products/productCategory/?0 [fecha de extracción: 05/05/2020]

[29] *2.0mm-5.0mm Thickness Hard Grey Paperboard for Furniture Liner*. (s. f.). New Bamboo Paper. Recuperado de: https://www.greypaperboard.com/2-0mm-5-0mm-thickness-hard-grey-paperboard-for-making-furniture-liner-board [fecha de extracción: 05/05/2020]

[30] González, J. (2015, 4 diciembre). *Construye tu pinza robótica porta acreditaciones*. DIWO. Recuperado de: http://diwo.bq.com/builds/pinza-robotica-porta-acreditaciones/ [fecha de extracción: 10/05/2020]

[31] SG92R (s. f.) Tower Pro Online Shop. Recuperado de: http://www.towerpro.com.tw/product/sg92r-7/ [fecha de extracción 05/05/2020]

[32] RS PRO. (s. f.). *RS Pro datasheet* [documento PDF]. Recuperado de: https://docs.rs-online.com/c133/0900766b8157a6fa.pdf [fecha de extracción 15/05/2020]

[33] Arduino. (s. f.). *MeArm Robot Arm - Your Robot - V1.0*. Arduino Project Hub. Recuperado de: https://create.arduino.cc/projecthub/benbobgray/mearm-robot-arm-your-robot-v1-0-326702 [fecha de extracción 05/04/2020]

[34] Llamas, L. (2016, agosto 23). *Motores paso a paso con Arduino y Driver A4988 o DR8825*. Luis Llamas. Recuperado de: https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/ [fecha de extracción 27/05/2020]

[35] Kysan Electronics (s. f.). *HB Stepper motors* [documento PDF]. Recuperado de: http://www.kysanelectronics.com/Catalog/HB_Step_Motors.pdf [fecha de extracción 25/05/2020]

[36] Robotdigg (s. f.). *Hybrid Stepping Motor 17HS6002-N27B* [documento PDF]. Recuperado de: http://www.robotdigg.com/upload/pdf/17HS6002-N27BA.pdf

[37] *Shop*. (2020, 6 julio). ProtoSupplies. Recuperado de: https://protosupplies.com/product/servo-motor-micro-sg92r/ [fecha de extracción 25/05/2020]

[38] Texas Instruments (2012, febrero). *DRV8834 Dual-Bridge Stepper or DC Motor Driver* [documento PDF]. Recuperado de: https://www.ti.com/lit/ds/symlink/drv8834.pdf [fecha de extracción 26/05/2020]

12. Apéndices

Anexo A. Objetivos de Desarrollo Sostenible

Uno de los objetivos del proyecto es la inclusión de al menos uno de los diecisiete Objetivos de Desarrollo Sostenible (ODS) de Naciones Unidas. Estos objetivos, pactados en 2015 con el fin de ser alcanzados en 2030, tienen un carácter global e intentan asegurar el bienestar de todos los ciudadanos del mundo.

Desde el principio, el proyecto estaba encaminado a lograr dos de estos objetivos: el objetivo número 10, "*Reducir la desigualdad en y entre los países*", y el objetivo número 13, "*Adoptar medidas urgentes para combatir el cambio climático y sus efectos*".

El proyecto tiene como objetivo principal el diseño de un robot que pueda simular ciertas discapacidades. El robot será controlado por el usuario como avatar y el cual deberá realizar diferentes pruebas. En caso de que el robot este simulando una discapacidad, el usuario sentirá de primera mano la impotencia, que servirá como concienciación y propulsor de un sentimiento de empatía.

El robot y las pruebas a las que se le quiere someter, deberán complicar la situación, pero nunca hacerla imposible. Se busca mentalizar a las personas con la idea de que "*discapacidad no significa incapacidad*". A partir de esta concienciación, se espera que las personas participantes adopten un comportamiento más inclusivo y empático.

En segundo lugar, el proyecto ha buscado sustituir los materiales más contaminantes por otros más ecológicos, como son la sustitución del plástico por el aluminio o el cartón. Estos últimos son materiales completamente reciclables o biodegradables cuyo impacto al medio ambiente es no es perjudicial. También se ha buscado sustituir la fuente de alimentación por un método reutilizable y menos dañino al medio ambiente que las pilas alcalinas.

El cartón es un material completamente reciclable y biodegradable. Aunque sus características mecánicas no sean las mismas que las del plástico, el realizar un robot

resistente a partir de materiales ecológicos no es imposible. El robot Zuri es un ejemplo de este avance, el cual salió al mercado hacer años con el carón gris como principal componente.

En conclusión, el proyecto se alinea con dos de los objetivos ODS propuestos por las Naciones Unidas, ya que pretende el bienestar y la reducción de las desigualdades buscando desarrollar un acercamiento emocional entre las personas. Y, por último, el diseño propone varias alternativas al plástico, que se suele utilizar en los robots con fin recreativo.

Anexo B. Código de Arduino para ESP32-CAM

Documento "ControlESP32CAM.ino"

```
#include "esp_camera.h"
#include <WiFi.h>
#include "pins_y_HTML.h"

const char* usuario_wifi = "MOVISTAR_ED6F";
const char* clave = "T6RpmA6Hoe96XGA8oSeC";

// Constantes de motor paso a paso
const int pin_direccion= 8;
const int pin_paso=9;
const int pasos_totales= 200;
//Constantes del servo
const int pin_servo = 2;

void startCameraServer();
void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config; //Estructura definida en la libreria esp_camera
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
```

```
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;//lower means higher quality
    config.fb_count = 2; //double speed
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

  // camera init
  //This function detects and configures camera over I2C interface,
  //allocates framebuffer and DMA buffers,
  //initializes parallel I2S input, and sets up DMA descriptors.
  //return ESP_OK on success
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Error 0x%x", err);
    return;
  }
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_QVGA);

  //Inicializacion de los motores
  //Pines de salida para el motor
  pinMode(pin_direccion, OUTPUT);
  pinMode(pin_paso, OUTPUT);
  //Pines de salida para el servo
  ledcSetup(8,50,8); //Canal 8, frecuencia de 50, resolucion de 8 bits
  // Al poner una resolucion de 8 bits, 180° sera 255
  ledcAttachPin(pin_servo, 8);

  // Comienza la conexion Wifi
  WiFi.begin(usuario_wifi, clave);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("Modulo conectado.");

  startCameraServer();

  Serial.print("Modulo listo! Use el siguiente link 'http://");
  Serial.print(WiFi.localIP());
  Serial.print("' ");
}

void loop() {
  delay(10000);
}
```

Documento "FuncionesServidorWeb.cpp"

```
#include "esp_http_server.h"
#include "esp_camera.h"
#include "Arduino.h"
#include "pins_y_HTML.h"

// Constantes de motor paso a paso
const int pin_direccion= 8;
const int pin_paso=9;
const int pasos_totales= 200;
//Constantes del servo
const int pin_servo = 2;

#define PART_BOUNDARY "123456789000000000000987654321"
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";

//HTTP Server Instance Handle. Every instance of the server will have a unique
handle.
httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }

    while(true){
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println("Camera capture failed");
            res = ESP_FAIL;
        } else {
             _jpg_buf_len = fb->len;
             _jpg_buf = fb->buf;
        }
        if(res == ESP_OK){
            size_t hlen = snprintf((char *)part_buf, 64, "Content-Type:
image/jpeg\r\nContent-Length: %u\r\n\r\n", _jpg_buf_len);
            res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
        }
        if(res == ESP_OK){
            res = httpd_resp_send_chunk(req, (const char *)_jpg_buf,
_jpg_buf_len);
```

```
    }
        if(res == ESP_OK){
            res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY,
strlen(_STREAM_BOUNDARY));
        }
        if(fb){
            esp_camera_fb_return(fb);
            fb = NULL;
            _jpg_buf = NULL;
        } else if(_jpg_buf){
            free(_jpg_buf);
            _jpg_buf = NULL;
        }
        if(res != ESP_OK){
            break;
        }
    }
    return res;
}

// CONTROL
static esp_err_t cmd_handler(httpd_req_t *req){
    char*  buf;
    size_t buf_len;
    char variable[32] = {0,};
    char value[32] = {0,};

    buf_len = httpd_req_get_url_query_len(req) + 1;
    if (buf_len > 1) {
        buf = (char*)malloc(buf_len);
        if(!buf){
            httpd_resp_send_500(req);
            return ESP_FAIL;
        }
        if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
            if (httpd_query_key_value(buf, "var", variable, sizeof(variable)) ==
ESP_OK &&  httpd_query_key_value(buf, "val", value, sizeof(value)) == ESP_OK) {
            } else {
                free(buf);
                httpd_resp_send_404(req);
                return ESP_FAIL;
            }
        } else {
            free(buf);
            httpd_resp_send_404(req);
            return ESP_FAIL;
        }
        free(buf);
    } else {
        httpd_resp_send_404(req);
        return ESP_FAIL;
    }
    int val = atoi(value);
    sensor_t * s = esp_camera_sensor_get();
```

```
int res = 0;
    // Control de las opciones de imagen
    if(!strcmp(variable, "framesize")) {
       res = s->set_framesize(s, (framesize_t)val);
    }else if(!strcmp(variable, "special_effect")){
      res = s->set_special_effect(s, val);
    }
    // Control remoto de coche
    // Control del servo
    else if(!strcmp(variable, "servo")){
      ledcWrite(8, val); // canal 8, dutycycle val
      Serial.println("Mi servo se mueve a posicion");
      Serial.println(val);
    }
    //Control del motor paso a paso. Empieza arriba en 0.
    else if(!strcmp(variable, "stepper")){
      if(val ==1){
        Serial.println("Arriba");
        digitalWrite(pin_direccion, HIGH);
        digitalWrite(pin_paso, HIGH);
      }else if (val==2){
        Serial.println("Abajo");
        digitalWrite(pin_direccion, LOW);
        digitalWrite(pin_paso, HIGH);
      }else if (val == 3){
        Serial.println("Stop");
        digitalWrite(pin_paso, LOW);
      }
    }else {
        res = -1;
    }
    if(res){
        return httpd_resp_send_500(req);
    }
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    return httpd_resp_send(req, NULL, 0);
}

// ESTADOS
static esp_err_t status_handler(httpd_req_t *req){
    static char json_response[1024];
    sensor_t * s = esp_camera_sensor_get();
    char * p = json_response;
    *p++ = '{';
    p+=sprintf(p, "\"framesize\":%u,", s->status.framesize);
    p+=sprintf(p, "\"special_effect\":%u,", s->status.special_effect);
    *p++ = '}';
    *p++ = 0;
    httpd_resp_set_type(req, "application/json");
    return httpd_resp_send(req, json_response, strlen(json_response));
}

// HTML
static esp_err_t index_handler(httpd_req_t *req){
```

```
httpd_resp_set_type(req, "text/html");
    httpd_resp_set_hdr(req, "Content-Encoding", "gzip");
    return httpd_resp_send(req, (const char *)mi_HTML, mi_HTML_len);
}

//HTTP
void startCameraServer(){
    //Definicion de todos los URI (identificadores)
    httpd_uri_t index_uri = {
        .uri       = "/",
        .method    = HTTP_GET,
        .handler   = index_handler,
        .user_ctx  = NULL
    };

    httpd_uri_t status_uri = {
        .uri       = "/status",
        .method    = HTTP_GET,
        .handler   = status_handler,
        .user_ctx  = NULL
    };

    httpd_uri_t cmd_uri = {
        .uri       = "/control",
        .method    = HTTP_GET,
        .handler   = cmd_handler,
        .user_ctx  = NULL
    };

   httpd_uri_t stream_uri = {
        .uri       = "/stream",
        .method    = HTTP_GET,
        .handler   = stream_handler,
        .user_ctx  = NULL
    };

    //  Generacion de una configuracion por defecto
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    // estructura definida en la libreria esp_http_server

    Serial.printf("Starting web server on port: '%d'\n", config.server_port);
    /* httpd_start crea una instancia de servidor web basado en los
     * requerimientos de la configuracion por defecto
     *  genera un handle a la instancia del servidor */
    if (httpd_start(&camera_httpd, &config) == ESP_OK) {
      //registra los URI handlers
        httpd_register_uri_handler(camera_httpd, &index_uri);
        httpd_register_uri_handler(camera_httpd, &cmd_uri);
        httpd_register_uri_handler(camera_httpd, &status_uri);
    }


config.server_port += 1;
    config.ctrl_port += 1;
```

```
Serial.printf("Starting stream server on port: '%d'\n", config.server_port);
    /* httpd_start crea una instancia de servidor web basado en los
     * requerimientos de la configuracion por defecto
     *  genera un handle a la instancia del servidor */
    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        //registra los URI handlers
        httpd_register_uri_handler(stream_httpd, &stream_uri);
    }
}
```

Documento "pins_y_HTML.h"

```
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

#define mi_HTML_len 2787
const uint8_t mi_HTML[] = {
0x1f,0x8b,0x08,0x00,0x38,0x9a,0xda,0x5e,0x00,0xff,0xb5,0x5a,0xfb,0x6f,0xe2,0x46,0
x1e,0xff,0x9d,0xbf,0x62,0xe2,0xdb,0x5b,0x48,0x97,0x77,0xb2,0xd9,0x2c,0x0b,0x54,0x
d9,0x34,0xb9,0x56,0xd7,0x4a,0xbd,0xa6,0xaa,0x2a,0x55,0xd5,0xed,0x60,0x0f,0xe0,0xc
6,0xd8,0xbe,0xb1,0x4d,0x48,0x53,0xfe,0xf7,0x7e,0xbe,0x33,0xe3,0xc7,0x18,0x93,0x4d
,0x2b,0x1d,0x88,0xc7,0x3c,0xbe,0xef,0xe7,0x0c,0x4c,0x4f,0xbc,0xc8,0x4d,0x1f,0x63,
0xc1,0xd6,0xe9,0x26,0x98,0xb7,0xa6,0xf9,0x87,0xe0,0x1e,0x3e,0x36,0x22,0xe5,0x2c,0
xe4,0x1b,0x31,0x6b,0x6f,0x7d,0xf1,0x10,0x47,0x32,0x6d,0xb3,0xfc,0xe1,0x46,0x61,0x
2a,0xc2,0x74,0xd6,0x7e,0xf0,0xbd,0x74,0x3d,0xf3,0xc4,0xd6,0x77,0x45,0x4f,0x0d,0xb
a,0xc5,0x26,0x3f,0xf4,0x53,0x9f,0x07,0xbd,0xc4,0xe5,0x81,0x98,0x8d,0xda,0x6c,0x00
,0xb4,0xa9,0x9f,0x06,0x62,0x7e,0x0d,0x78,0x19,0x05,0x6c,0xe9,0x87,0x3c,0x60,0x37,
0x77,0xdf,0x9f,0x8d,0x7b,0xd7,0x57,0xdf,0x4d,0x07,0x7a,0xb9,0x35,0x4d,0xd2,0x47,0
x7c,0xb2,0x56,0x6b,0x11,0x79,0x8f,0xec,0xa9,0xc5,0x98,0x42,0x3e,0x19,0x0d,0x87,0x
ff,0xfc,0x80,0xd1,0x12,0x08,0x7a,0x4b,0xbe,0xf1,0x83,0xc7,0x09,0xbb,0x92,0x20,0x4
3,0xb3,0x0b,0xee,0xde,0xaf,0x64,0x94,0x85,0xde,0x84,0xfd,0x63,0x74,0x49,0x4f,0x9a
,0x76,0xa3,0x20,0x92,0x98,0xb9,0xb9,0xa5,0x67,0x01,0x9e,0xf8,0xbf,0x8b,0x09,0x1b,
0x5d,0xc4,0xbb,0xd6,0x5e,0x73,0xdc,0x4a,0x84,0x9b,0xfa,0x51,0xd8,0xdf,0x70,0x3f,0
x54,0x54,0x3d,0x3f,0x89,0x03,0x0e,0x1a,0xcb,0x40,0x60,0x5b,0x6b,0xf0,0xc5,0x4d,0x
92,0xfa,0x41,0xc4,0x3c,0xc1,0x02,0xce,0xfc,0x0d,0x5f,0x89,0xf0,0x8b,0x41,0x6b,0xe
9,0xaf,0x32,0x29,0x08,0x62,0x2d,0xfc,0xd5,0x3a,0x9d,0xb0,0xf3,0xf7,0xdb,0x35,0x91
,0x8a,0xb9,0xe7,0xf9,0xe1,0x6a,0x32,0xa4,0xc1,0x86,0xcb,0x95,0x1f,0xea,0xef,0xbd,
0x07,0xb1,0xb8,0xf7,0xd3,0x9e,0x9e,0xeb,0x2d,0xc4,0x32,0x92,0xa2,0xba,0xad,0xb7,0
x08,0x22,0xf7,0xbe,0x97,0xa4,0x5c,0xa6,0x8d,0x20,0x7c,0x99,0x0a,0xd9,0x00,0x21,0x
20,0x7f,0xd3,0xfe,0x0a,0x26,0x33,0xe3,0x87,0x81,0x1f,0x8a,0x67,0x48,0x14,0xa8,0x6
c,0x00,0x35,0xcd,0xa0,0x0f,0x2d,0x37,0xf4,0xb0,0xb2,0xb4,0xa5,0xf8,0x20,0x30,0x6d
,0x35,0x76,0x3e,0x1c,0xc6,0x3b,0x1a,0xd7,0xb5,0x63,0xd0,0xa6,0x51,0x0c,0x4b,0x98,
0x3d,0x8b,0x48,0x7a,0x42,0xf6,0x24,0xf7,0xfc,0x2c,0xc1,0x56,0x9a,0xdd,0xb7,0xfa,0
x4a,0xd7,0x3d,0xf2,0x3c,0x18,0x47,0x48,0x65,0x9e,0x38,0x4a,0xe0,0x63,0x51,0x38,0x
61,0x52,0x04,0x3c,0xf5,0xb7,0x42,0x21,0x05,0x46,0x43,0x78,0x74,0x01,0xa4,0x80,0x3
6,0x86,0x65,0x29,0x5f,0x04,0xca,0x4c,0x7a,0xbd,0xe0,0xcb,0xf2,0x9c,0xb3,0x0b,0x7a
,0x1e,0x63,0xa5,0x99,0xe9,0x8a,0x99,0xf7,0xad,0x45,0x96,0xa6,0x20,0xd6,0xac,0x12,
0xe3,0x03,0x6f,0x35,0x5c,0xe1,0xd6,0x96,0x7e,0xde,0xd7,0xb0,0x32,0x4b,0x35,0x18,0
x2a,0xb7,0xce,0x64,0x42,0x7e,0x1d,0x47,0x3e,0xa2,0x51,0x36,0x7b,0xba,0x25,0xd8,0x
```

```
d5,0xbb,0xf7,0x57,0xb7,0x7a,0xda,0x16,0xcc,0x30,0x53,0x0b,0x0b,0x9a,0x8a,0xb2,0x9
4,0x2c,0x0e,0x92,0x85,0x5c,0x13,0x0e,0x5d,0x6e,0x85,0x12,0xcf,0x42,0x7f,0x3d,0xfa
,0x78,0x79,0x7b,0x5b,0xec,0xeb,0xc7,0xe2,0x7f,0x59,0x35,0x26,0xce,0x0c,0x4e,0x63,
0x9b,0x31,0xd9,0x46,0x99,0x36,0x79,0xf0,0x53,0x77,0x7d,0x44,0x5f,0xcd,0x36,0x56,0
x6e,0x98,0x7b,0xd3,0x78,0x7c,0x94,0xff,0xea,0x96,0x0a,0x29,0x3f,0x8c,0xb3,0x54,0x
11,0x2c,0x05,0x54,0xe2,0xc6,0xdc,0xf5,0x53,0x90,0x57,0x23,0xc3,0xa8,0xfa,0x9e,0x2
3,0x22,0x3d,0xf4,0x93,0xc0,0x87,0xfa,0x2a,0x79,0x89,0xbd,0xad,0x39,0x78,0xce,0x52
,0x4d,0xd1,0xf9,0x74,0x83,0xed,0x4a,0x55,0xc2,0xc9,0x95,0x19,0x57,0x52,0x3c,0x96,
0xd4,0xba,0xe6,0x73,0xa2,0x33,0x85,0xad,0x2d,0x13,0x97,0x85,0xd2,0x52,0xc9,0xc3,0
x5c,0x6d,0xfd,0xf3,0xa4,0xc4,0x52,0x85,0x6e,0xd6,0xac,0xc9,0xed,0x13,0xe6,0x38,0x
4d,0x9e,0xa2,0xf3,0x6f,0x2e,0xe6,0xc8,0x72,0xe3,0x62,0x18,0x88,0x25,0x05,0xb9,0x1
e,0xa8,0x38,0xc9,0xd5,0x71,0x20,0xa4,0xf1,0x55,0x70,0xa8,0x6c,0x32,0x71,0xd7,0xc2
,0xbd,0x17,0xde,0x9b,0xaa,0x8e,0x1b,0xa0,0xb4,0x2b,0x1f,0x83,0xaa,0x4a,0x99,0x67,
0x34,0xa5,0x12,0xcc,0x6e,0x26,0x5a,0x3b,0x90,0x58,0xfc,0xdc,0x19,0xc3,0x4f,0x4e,0
x0b,0x8d,0x35,0x2f,0xab,0xfc,0x11,0x20,0x83,0x68,0x66,0x4c,0x0c,0x8e,0xe2,0x1d,0x
4b,0x22,0x90,0xab,0x26,0x3c,0xaa,0x07,0x1a,0xf1,0xed,0xc2,0x55,0xf7,0x0f,0xdb,0xf
d,0x0e,0xc3,0x32,0x97,0xf0,0x17,0xaa,0xd4,0x33,0x30,0xb6,0x12,0xbf,0x3e,0xb1,0x8a
,0x58,0x3c,0x8e,0x05,0xc7,0xbc,0x2b,0x26,0x61,0x14,0x2a,0x0b,0x36,0x52,0x1c,0x1b,
0x76,0x2a,0x11,0x5b,0xe1,0xdb,0x24,0x25,0x30,0x45,0x8c,0x34,0xd0,0x9c,0x2c,0x23,0
x37,0x4b,0xaa,0x01,0x43,0x81,0x70,0xb8,0x6d,0x92,0xf3,0xa5,0x0d,0xd1,0x93,0x59,0x
18,0x52,0xce,0x25,0xf5,0xbb,0xf7,0x87,0x85,0x3c,0xe7,0xce,0x0a,0x8b,0xc6,0xa8,0x9
8,0x54,0xb3,0x9a,0xa5,0x27,0x3b,0x39,0xa2,0x2e,0xb1,0xdc,0x30,0xb6,0x6f,0x3d,0xcb
,0x68,0xba,0xce,0x36,0x8b,0x8a,0x7d,0x4b,0xf3,0xca,0xd5,0x82,0x77,0x86,0xdd,0x61,
0xf7,0x0c,0x6f,0xca,0x57,0x72,0xa6,0x0d,0x61,0x2d,0x51,0x3e,0xb2,0x79,0xcb,0xf3,0
x42,0x55,0x8e,0x32,0x0d,0x1f,0xca,0xfb,0x8c,0x59,0x2b,0x65,0xa7,0xf7,0xbe,0xd9,0x
33,0xb4,0x95,0x5e,0x60,0x34,0x43,0xb5,0x1e,0x51,0xd2,0x26,0xfa,0x1d,0xa2,0x60,0xf
0,0x7f,0x35,0xa0,0x65,0xbe,0x97,0xd8,0xae,0xc2,0xd6,0xdf,0xb0,0x9b,0x61,0xd6,0xd8
,0xad,0x31,0x53,0x37,0xd9,0xad,0x0c,0x17,0x5b,0x4c,0xca,0xae,0x6b,0xdf,0xf3,0x44,
0xad,0xe4,0x93,0xe1,0xb0,0xd6,0x9a,0x0e,0x74,0x3b,0x8b,0x2f,0xba,0xb9,0x6e,0x4d,0
xa9,0xad,0xa5,0x4f,0xdd,0x3f,0xcd,0x01,0x35,0xf5,0xfc,0x2d,0xf3,0xbd,0x99,0x93,0x
a4,0x52,0xf0,0x4d,0xd9,0xe8,0x38,0xcc,0x0d,0x78,0x92,0xcc,0x9c,0x5a,0x03,0xe4,0xa
0,0x3d,0x06,0x18,0x5a,0xaf,0x0a,0x98,0xc3,0x12,0xe9,0xce,0x1c,0xac,0x4d,0x07,0x40
,0x48,0x24,0x73,0x12,0x68,0xaa,0x4d,0x13,0x44,0xdb,0x75,0x89,0x4e,0x9c,0xf9,0x54,
0xb5,0x44,0xb4,0x9c,0x4a,0x0c,0x3c,0xc6,0x03,0x7f,0x15,0xce,0x1c,0x4a,0xe1,0xce,0
xfc,0x75,0xb8,0x48,0xe2,0x0f,0xd5,0xf7,0xef,0xfd,0xf0,0x77,0x0e,0xec,0x29,0xe4,0x
a0,0x9e,0xb9,0x0e,0x42,0x3d,0x48,0x12,0x73,0x0c,0x47,0x40,0xae,0xeb,0xac,0xb2,0x9
d,0xa3,0xec,0xe5,0x68,0x66,0x85,0xdc,0x46,0x0e,0x35,0x6a,0x33,0x67,0x88,0x4f,0xbe
,0x9b,0x39,0xe3,0xb7,0x6f,0x1d,0xb6,0xe5,0x41,0x86,0xad,0x98,0x33,0x42,0x7b,0x62,
0xc9,0xb3,0x00,0xf1,0xa0,0x58,0x77,0x14,0x9f,0x05,0x41,0x17,0x75,0x8a,0x14,0x24,0
xa3,0x07,0x4d,0x72,0x0c,0x92,0xa6,0xf9,0x22,0x32,0x5c,0x4a,0x7f,0xc1,0x1d,0xc5,0x
67,0x14,0x1a,0x6b,0x79,0xd1,0x03,0x76,0xe2,0xec,0x93,0x6d,0x00,0xde,0x27,0x6b,0x7
1,0xb4,0x04,0x37,0x5b,0x8c,0x3a,0x1a,0x22,0x96,0x22,0x49,0x4e,0x3f,0x38,0xa4,0xe0
,0x0a,0x64,0x16,0x7f,0x06,0x0e,0x65,0x54,0xf0,0x44,0x1c,0x42,0x62,0x7a,0x0b,0xb1,
0x5e,0x08,0x3c,0xbf,0x52,0x13,0xd3,0x81,0x16,0x65,0xae,0x95,0xfd,0x52,0x03,0x5d,0
x63,0xdd,0xe3,0x9e,0x02,0xd2,0x6c,0x1c,0x00,0xa9,0x59,0xcc,0x9b,0xaa,0x46,0xaa,0x
5a,0x4a,0x1c,0xf5,0xa8,0x70,0x1d,0xd7,0x7c,0x7e,0xae,0x9b,0x46,0xb1,0x72,0x23,0x6
3,0xab,0x4b,0x67,0xfe,0xf3,0xbf,0xae,0x3a,0xa3,0xe1,0xf8,0x7c,0xf7,0xee,0xe2,0xf2
,0x74,0x3a,0xd0,0xeb,0x47,0x01,0xde,0x39,0xf3,0xbb,0x9f,0x00,0x71,0x39,0x1c,0xee,
0x2e,0x86,0xc3,0xcf,0x03,0x5c,0x38,0x73,0xda,0x7f,0x71,0x3e,0xdc,0x9d,0x5f,0xbe,0
x60,0x3f,0xfc,0x48,0x4b,0x26,0x94,0xab,0xe9,0x6f,0xce,0xfc,0xfa,0x9b,0xdb,0x0e,0x
1a,0xfe,0xdd,0xf8,0xfd,0xc5,0xe7,0x71,0x9c,0x3b,0xf3,0xff,0x10,0xd1,0xb3,0x31,0x0
0,0xce,0x5f,0x40,0xf4,0xcc,0x99,0x7f,0xad,0x20,0xb0,0x7b,0x37,0x7a,0xf7,0x02,0x12
,0x43,0x90,0x50,0x10,0x38,0xa4,0xec,0x46,0xe3,0x43,0x1a,0xc8,0x1b,0x8a,0x79,0xed,
0x00,0x78,0x93,0x85,0x17,0x48,0x04,0xf8,0x61,0x1c,0x14,0xa1,0x37,0x86,0x06,0x28,0
```

```
xe5,0xb0,0x59,0xbb,0xda,0x91,0xc4,0xbb,0xb6,0xe5,0x30,0x85,0x63,0x1d,0x8f,0x28,0x
18,0xcb,0x8a,0xa8,0x05,0xff,0x0d,0x81,0xfb,0x57,0x22,0x8a,0x20,0xfe,0x46,0x40,0x1
1,0xd8,0xdf,0x8d,0x27,0x1b,0x76,0x7e,0x45,0xe3,0xe6,0x68,0x82,0x46,0x1b,0xc3,0x8a
,0x1d,0xc6,0xd5,0x1d,0xae,0x04,0x54,0xf7,0xf9,0x5c,0x64,0xa9,0x2c,0x6e,0x02,0x48,
0x1f,0x35,0x8a,0x24,0xa8,0xf2,0x5e,0x2c,0x5c,0xdc,0x55,0xfc,0x57,0x2c,0x97,0xb0,0
xab,0x63,0x12,0xa3,0x6a,0x5e,0x17,0xd1,0xee,0x68,0xe8,0x4d,0x03,0xbe,0x10,0x41,0x
81,0x57,0x15,0x75,0x07,0xbd,0xa6,0x3c,0xc4,0x08,0xf1,0xd4,0x66,0x7c,0x52,0xfe,0x2
f,0xf3,0xb4,0xe1,0xd7,0x38,0x4f,0xd3,0xa3,0xae,0x09,0x49,0xc5,0xd2,0x72,0x2a,0xcb
,0x15,0xd2,0x68,0xb5,0x42,0x4b,0x91,0xd7,0x1d,0xc3,0x9c,0x3a,0xf3,0x21,0xc0,0xd1,
0x2d,0xa4,0x0c,0xef,0x58,0x2b,0x34,0xaf,0x7d,0x5a,0x6b,0x7f,0x60,0xca,0x0e,0xf9,0
xb8,0x92,0x12,0x93,0x48,0x48,0xae,0xf4,0xe3,0x14,0x1b,0x5b,0x83,0x01,0xbb,0xcd,0x
42,0xb5,0x92,0xe0,0x04,0xa1,0x4e,0x25,0x74,0x63,0xa4,0xc9,0xe1,0x4c,0x0c,0x79,0xf
4,0x61,0x25,0x49,0x19,0x3a,0x2f,0x5c,0x9d,0x41,0x45,0xb3,0xb9,0x2a,0xc1,0xf4,0x10
,0x41,0x5f,0xf1,0xf4,0xad,0x9f,0xa4,0x7d,0x9c,0xa3,0x3b,0x6d,0x5d,0xa3,0xdb,0xa7,
0xd8,0xb0,0x2f,0x60,0x93,0x75,0xf4,0x50,0x83,0xb5,0x20,0xa5,0xd8,0x44,0x5b,0x51,0
x03,0x2e,0xa0,0xe9,0x6a,0x0c,0xd0,0x85,0x23,0xae,0x44,0x7a,0x13,0x08,0xfa,0xfa,0x
f1,0xf1,0x1b,0x90,0xd4,0xec,0x2a,0xa8,0x12,0x80,0x6e,0xbf,0xf4,0x1d,0xc6,0x67,0x2
1,0xcb,0x72,0x5f,0xc1,0xa1,0x97,0x3e,0x6a,0x53,0x3c,0x83,0xc2,0xb2,0x90,0x05,0x1f
,0xc5,0xda,0x34,0x80,0xee,0x9c,0x96,0x82,0x3f,0xf8,0x21,0x62,0xb9,0x4f,0xeb,0x1d,
0xd5,0x30,0xa1,0x91,0xae,0xd0,0xea,0xfb,0x21,0x18,0xf9,0xfa,0xc7,0xef,0xbe,0x05,0
x5c,0xfb,0x8e,0x2e,0x8d,0x8c,0x89,0xdb,0xb6,0x4a,0x69,0xa5,0x91,0x00,0x49,0xdf,0x
47,0x87,0x82,0xe9,0x4f,0xaf,0x9e,0x0a,0xc6,0x71,0x6a,0xc5,0xf1,0x13,0x04,0x22,0xf
8,0x1c,0xc2,0xec,0x0d,0x6b,0x4f,0x2e,0x47,0xed,0x3d,0xfa,0x26,0x42,0xf2,0x49,0x73
,0x02,0x43,0x75,0x2c,0xf5,0x91,0x44,0xcf,0x73,0x18,0xc5,0x35,0x06,0xad,0xcd,0x51,
0xe8,0x06,0xbe,0x7b,0x5f,0xe3,0xb1,0xaa,0xe3,0x1b,0xd5,0x36,0x7b,0xd8,0x71,0x8c,0
xca,0xec,0x90,0x0e,0x6e,0x3e,0x97,0xac,0x63,0x21,0x38,0x2d,0xbc,0xb2,0xd4,0x7d,0x
47,0xb3,0xbf,0x87,0xbb,0x25,0xfa,0xa4,0xaa,0xd7,0x0b,0xdd,0xe5,0x1b,0xb4,0xc7,0x6
9,0xb6,0xb2,0xd8,0xc3,0xb9,0xf4,0x27,0x2a,0x20,0xc4,0xb6,0x08,0xba,0xba,0x9a,0x74
,0xcd,0xca,0x0f,0xf0,0xd6,0x54,0x54,0xa4,0xa9,0x4e,0x03,0xc2,0x1e,0xce,0x58,0x98,
0x05,0x01,0xfb,0x12,0x27,0x5e,0xe0,0x9b,0x58,0xab,0x0a,0x3a,0x10,0xc8,0x5b,0xfa,0
x1a,0x57,0xd1,0x2c,0xa4,0x43,0x88,0xa8,0xab,0x63,0x25,0x7f,0x9e,0xbf,0xda,0xa5,0x
98,0x55,0x20,0x15,0x5d,0x7d,0x73,0x42,0x37,0xeb,0x8a,0x69,0x2c,0x9c,0x9c,0xa8,0x6
f,0x95,0x90,0xd5,0xdb,0xb0,0x54,0x2e,0xd4,0x54,0x74,0x88,0xbb,0x86,0x23,0x47,0x5e
,0xc1,0x50,0x30,0x6e,0x29,0xe0,0xf5,0x6b,0x1b,0xd9,0x09,0xa4,0x51,0x40,0xa5,0x20,
0x7a,0x3f,0x5c,0x0e,0x6d,0x34,0xa4,0x36,0x51,0xb1,0xa7,0xdc,0x03,0xa3,0x2c,0x4d,0
x8a,0xb2,0xb6,0x91,0x76,0x72,0x04,0xa4,0xc0,0x92,0x0d,0x73,0xf9,0x94,0x6b,0xaf,0x
a4,0xe2,0xa2,0x52,0x55,0xf4,0x38,0x29,0x5a,0x86,0x5c,0x94,0x8a,0x66,0xbe,0x64,0x6
3,0x98,0x6a,0x58,0x6c,0x59,0xc0,0x57,0xee,0x2d,0x3c,0xaa,0xd1,0x2e,0x90,0xe8,0x39
,0xdd,0x49,0xf4,0x70,0x22,0x69,0xc6,0x5e,0x55,0x61,0x13,0x4e,0x9d,0xc2,0xeb,0x48,
0xb3,0xc5,0xc6,0x4f,0x1b,0x10,0xb6,0x47,0x3a,0x10,0xea,0xb8,0x4c,0x61,0x2b,0x01,0
xa4,0x48,0x33,0x19,0x1a,0x1b,0x55,0xc2,0x0f,0x45,0x44,0x3e,0x3e,0x9f,0x25,0xf6,0x
03,0x53,0x13,0xbe,0xdc,0x72,0x39,0x7b,0xf5,0x04,0x21,0x7c,0x6f,0xff,0x1a,0x2c,0x6
0,0xa0,0x18,0xd9,0x7f,0xd2,0x28,0x97,0x02,0x5a,0xef,0x28,0x94,0x3a,0xa4,0x18,0xeb
,0xa7,0x6b,0x11,0x76,0xd0,0x95,0xc4,0x20,0x07,0x8e,0xcb,0xb2,0xa1,0x39,0x88,0x02,
0x01,0x7a,0xab,0xce,0x27,0x49,0xf5,0x0c,0x0c,0xa1,0x04,0xbd,0x7a,0x52,0x28,0xf6,0
xf4,0xab,0x85,0x9f,0xac,0x85,0xd7,0xa5,0x68,0x4d,0xe9,0xa2,0xe6,0xd5,0x53,0x8e,0x
0a,0xc9,0x93,0xa6,0xf6,0x9f,0x72,0x4a,0xfb,0xbc,0xdc,0xa0,0xa2,0x7d,0x2b,0x90,0x
3,0xc1,0x1a,0x9a,0x93,0x04,0x9e,0x47,0x85,0x1b,0x5f,0x1e,0xa1,0x85,0x24,0x26,0x2a
,0x3c,0xa1,0x2b,0x24,0xc5,0xed,0xb3,0x82,0x6b,0x1a,0x86,0x84,0x16,0xa5,0xf0,0xc4,
0x42,0xa8,0xd2,0xbb,0xb4,0x92,0x15,0x15,0xc5,0xe2,0x6f,0x49,0x14,0xe6,0xc9,0xa5,0
x19,0x07,0x11,0xa8,0x20,0xc8,0x39,0x29,0x54,0xd4,0x57,0x9a,0xb8,0x53,0x3e,0x15,0x
c9,0xab,0x20,0xe8,0xb4,0xfb,0x76,0xd3,0xa2,0xca,0x8d,0xd9,0x8c,0x46,0xe5,0x86,0x4
3,0x26,0xbb,0x3e,0x97,0xd1,0xa5,0x82,0x4f,0xe5,0x32,0x45,0xf7,0x17,0x65,0xc9,0x5f
,0xbb,0x6c,0xc9,0x11,0xf6,0x25,0x1e,0xc3,0xab,0xfe,0x80,0x36,0x01,0x16,0x49,0x5c,
```

```
0x62,0x4a,0x72,0x2b,0x62,0x84,0xce,0xde,0x55,0x4e,0x5f,0xcc,0x65,0x9d,0x43,0x0a,0
x08,0x14,0x87,0x35,0xc5,0x51,0x51,0x1d,0xea,0x99,0x80,0x18,0xa1,0x26,0x44,0x9b,0x
b6,0x6c,0x56,0x80,0x8b,0x99,0xdf,0xb7,0x12,0x14,0x3c,0xce,0xfe,0x2d,0x1e,0x55,0xf
f,0x8b,0x8d,0xf0,0x54,0x76,0x2f,0x1e,0xf5,0x81,0x0f,0x98,0xd5,0xc5,0x87,0x99,0x5c
,0xa8,0xb6,0x55,0x4d,0x52,0x76,0x01,0xce,0x9b,0x4d,0x46,0xb7,0x90,0x05,0x02,0xd4,
0xe7,0x74,0xcd,0x7e,0x8c,0x32,0x77,0x6d,0xc0,0x2a,0xc7,0x55,0x00,0x86,0xe8,0x45,0
xb0,0x77,0x11,0x71,0xe9,0xe9,0x4e,0xb8,0x0d,0xbc,0xd4,0x9d,0xb7,0xbb,0xec,0x89,0x
be,0x5f,0x47,0x1e,0xe2,0xff,0xec,0xb2,0xcb,0xda,0x0f,0x6b,0xdf,0x5d,0xd3,0xf7,0xb
d,0xca,0x69,0x25,0x36,0xd3,0x38,0x1f,0xc5,0x97,0xc5,0x2f,0xc4,0x56,0x34,0xfe,0x7
,0x81,0xb5,0xf3,0x61,0x89,0xec,0x7c,0x58,0x43,0xf6,0x57,0x39,0x3b,0x44,0xa6,0xf4,
0x5a,0xa8,0x53,0x41,0x92,0x59,0xca,0x16,0x0a,0x7d,0xa2,0x9a,0xa5,0xd6,0x4f,0xa0,0
xc8,0x57,0xd8,0xcc,0xa3,0xa3,0x83,0x99,0x28,0x3c,0xd5,0x6e,0xac,0xbe,0xf7,0x81,0x
3e,0x30,0xbe,0xd2,0xbe,0x95,0xb7,0x4e,0xba,0xe0,0x30,0xbd,0xbf,0x6f,0xb8,0xa2,0xa
2,0xdb,0x3e,0xbb,0x44,0xa9,0x44,0xf1,0xe9,0x9c,0x14,0x76,0x2f,0xc7,0xda,0x0a,0xa7
,0x88,0xbe,0xaa,0xa7,0x8c,0x3e,0xe8,0x1a,0x46,0x05,0x0e,0x2a,0x20,0xcc,0x0d,0x88,
0xcf,0x87,0x15,0xc4,0x06,0x51,0x9d,0xd0,0x29,0xe1,0x2d,0x9c,0x2d,0xc7,0xab,0x55,0
x6d,0x8e,0x09,0x4a,0x47,0x3f,0x18,0x6d,0x97,0x5a,0xfa,0x8c,0x9a,0x48,0xfd,0x96,0x
92,0x96,0x4b,0xa3,0x25,0xad,0x09,0x33,0x77,0xa8,0x89,0x3f,0xfe,0xd0,0x4a,0xaa,0xa
d,0x91,0x30,0x35,0x2d,0x0c,0x3f,0xd8,0x81,0x62,0xf3,0x3e,0x18,0xdc,0xe1,0x87,0x49
,0x04,0xdf,0x66,0xc3,0x43,0x4f,0x1d,0x1b,0xee,0x5c,0x5c,0x57,0xf7,0xd9,0x57,0x22,
0xc6,0x8a,0x8f,0x57,0xfe,0xcb,0xad,0xe7,0x4b,0xe1,0xba,0x94,0xec,0x38,0xf6,0xe1,0
x6a,0xb5,0x8b,0x5b,0xa8,0xd0,0xe3,0x18,0xba,0xf0,0xb7,0x2c,0xe4,0xe4,0x77,0x3e,0x
75,0x6f,0x2c,0x63,0x11,0x8e,0x33,0xba,0x81,0xc5,0x1a,0x2e,0xff,0x24,0x64,0x77,0x3
5,0x99,0x99,0x8a,0x61,0xbd,0x12,0x05,0x5e,0x6d,0xd6,0x6a,0xab,0x45,0xfa,0x0d,0x1d
,0x5e,0x51,0x00,0x8a,0x54,0xdb,0xb1,0x34,0x54,0x31,0xda,0x93,0x4d,0x45,0x5b,0x4a,
0x49,0x98,0xb2,0xaf,0x88,0x77,0x95,0xa8,0x6f,0x23,0xf9,0x80,0x10,0xa8,0x7b,0x46,0
xc5,0xab,0x0e,0xf1,0x8c,0x15,0x1e,0x66,0x23,0xfa,0x88,0x6b,0xce,0x3a,0xa6,0x43,0x
d0,0xb7,0x06,0x14,0x3d,0xac,0xd9,0x58,0x61,0xbe,0xb6,0xfb,0x64,0x56,0x51,0x47,0x2
1,0x65,0x5e,0xde,0x8e,0xd5,0xb6,0x37,0x6d,0xab,0xaa,0xc3,0xbb,0x70,0x0f,0x2e,0x55
,0x51,0x6f,0xbf,0xb1,0x29,0x14,0x51,0x86,0x7b,0x81,0x82,0x10,0x98,0xb4,0x77,0xc1,
0x45,0xb4,0x8f,0x74,0xf1,0x53,0xad,0xbe,0x05,0xce,0xcf,0xc2,0x03,0x73,0xda,0x34,0
xa3,0xfc,0x4a,0x76,0xa0,0xfe,0x07,0xf1,0x27,0x0d,0x9e,0xfa,0x67,0x1e,0x21,0x00,0x
00
};
```

Anexo C. Documento HTML para ESP32-CAM.

```
<!doctype html>
<html>
<head>
<meta name='viewport'        content='width=device-width,        initial-
scale=1' />
<title>Control final ESP32-CAM</title>
<style>

body {
  width:100%;
  font-family: Arial;
  background: #181818;
  color: #EFEFEF;
  font-size: 16px
}
section.main {
  display: flex
}
/*Estilo de la imagen*/
figure{
  height: 49vh;
  padding:0;
  margin:0;
  -webkit-margin-before:0;
  margin-block-start:0;
  -webkit-margin-after:0;
  margin-block-end:0;
  -webkit-margin-start:0;
  margin-inline-start:0;
  -webkit-margin-end:0;
  margin-inline-end:0
}
figure img{
  display: block;
  width: 400px;
  height: 49vh;
  margin-top: 10px;
  border-radius: 4px;
}
.image-container {
  position: relative;
  min-width: 160px
}
section table{
  width:400px;
  background: #363636;
  border-radius: 4px;
  margin-top: 10px;
  padding:0;
}
button {
```

```
  display: block;
  margin:5px;
  width:100px;
  height:90px;
  padding: 0px;
  border: 0;
  cursor: pointer;
  color: #EFEFEF;
  background: #A79AFF;
  border-radius: 5px;
  font-size: 16px;
  outline: 0
}
button:active {
  background: #C1B8FF
}
button.peque{
  height:36px;
  width: 260px;
}
.switch {
  display: block;
  position: relative;
  line-height: 22px;
  font-size: 16px;
  height: 22px
}
.switch input {
  outline: 0;
  opacity: 0;
  width: 0;
  height: 0
}
.slider {
  width: 50px;
  height: 22px;
  border-radius: 22px;
  cursor: pointer;
  background-color: grey
}
.slider,.slider:before {
  display: inline-block;
  transition: .4s
}
.slider:before {
  position: relative;
  content: "";
  border-radius: 50%;
  height: 15px;
  width: 15px;
  left: 4px;
  top: 2px;
  background-color: #EFEFEF
}
```

```
input:checked+.slider {
  background-color: #A79AFF
}
input:checked+.slider:before {
  -webkit-transform: translateX(26px);
  transform: translateX(26px)
}
select {
  border: 1px solid #363636;
  font-size: 14px;
  width:100%;
  height: 22px;
  outline: 0;
  border-radius: 5px
}
input[type=range]{
  -webkit-appearance:none;
  width:100%;
  height:24px;
  background:#363636;
  margin:2px 0;
}
input[type=range]:focus{
  outline:0
}
input[type=range]::-webkit-slider-runnable-track{
  width:100%;
  height:2px;
  cursor:pointer;
  background:#EFEFEF;
  border-radius:0px;
  border:0
  solid #EFEFEF
}
input[type=range]::-webkit-slider-thumb{
  border:1px solid rgba(0,0,30,0);
  height:20px;
  width:20px;
  border-radius:50px;
  background:#A79AFF;
  cursor:pointer;
  -webkit-appearance:none;
  margin-top:-9px
}
input[type=range]:focus::-webkit-slider-runnable-track{
  background:#EFEFEF
}
input[type=range]::-moz-range-track{
  width:100%;
  height:2px;
  cursor:pointer;
  background:#EFEFEF;
  border-radius:0;
  border:0 solid #EFEFEF
```

```
}
input[type=range]::-moz-range-thumb{
  border:1px solid rgba(0,0,30,0);
  height:22px;
  width:22px;
  border-radius:50px;
  background:#363636;
  cursor:pointer
}
.hidden {
  display: ncne
}

</style>
</head>

<body>
<figure>
  <div id="stream-container" class="image-container">
  <img id="stream" src=""> </div>
</figure>

<section id="buttons"><table>

<tr><td align="left">   Pinza </td>
    <td align="left" colspan="1"><input type="range" id="servo" min="0" max="255"
value="0" class="default-action">

<td align="center" rowspan="2"><button id="arriba"
    onpointerdown="document.dispatchEvent(arribapress);"
    onpointerup="document.dispatchEvent(arribarelease);"
    onpointerleave="document.dispatchEvent(arribarelease);">Arriba</button></td>

<tr><td align="left">   Calidad</td>
    <td align="left">
      <select id="framesize" class="default-action">
        <option value="8">XGA(1024x768)</option>
        <option value="7">SVGA(800x600)</option>
        <option value="6">VGA(640x480)</option>
        <option value="5" selected="selected">CIF(400x296)</option>
        <option value="4">QVGA(320x240)</option>
        <option value="3">HQVGA(240x176)</option>
        <option value="0">QQVGA(160x120)</option>
      </select></td></tr>

<tr><tr> <td align="center" colspan="2" style ='font-size: 1px'>  </td>

<td align="center" rowspan="7"><button id="abajo"
    onpointerdown="document.dispatchEvent(abajopress);"
    onpointerup="document.dispatchEvent(abajorelease);"
    onpointerleave="document.dispatchEvent(abajorelease);">Abajo</button></td>

<tr><tr>
<tr><td align="left">    Color</td>
```

```
    <td align="left"><div class="switch"><input id="special_effect"
type="checkbox" class="default-action"><label class="slider" for="special_effect"
></label></div></td>

    <tr><tr>
<tr><td align="right" colspan="2"><button id="toggle-stream" class="peque">Start
Stream</button></td>

</table>
</section>

 <script>
// Funciones para el streaming
  const hide = el => {
      el.classList.add('hidden')
  }
  const show = el => {
    el.classList.remove('hidden')
  }

  const view = document.getElementById('stream')
  const imagen = document.getElementById('stream-container')
  const BotonDeStream= document.getElementById('toggle-stream')

  const stopStream = () => {
    window.stop();
    BotonDeStream.innerHTML = 'Start Stream'
  }
  const startStream = () => {
    view.src = `${document.location.origin + ':81'}/stream`
    show(imagen)
    BotonDeStream.innerHTML = 'Stop Stream'
  }
  BotonDeStream.onclick = () => {
    const BotonActivado = BotonDeStream.innerHTML === 'Stop Stream'
    if (BotonActivado) {
      stopStream()
    } else {
      startStream()
    }
  }

// Funciones para el control
const updateValue = (el, value, updateRemote) => {
    updateRemote = updateRemote == null ? true : updateRemote
    let initialValue

    if (el.type === 'checkbox') {
      initialValue = el.checked
      value = !!value
      el.checked = value
    } else {
      initialValue = el.value
      el.value = value
```

```
    }
    if (updateRemote && initialValue !== value) {
      updateConfig(el);
    }
  }

function updateConfig (el) {
    let value
    switch (el.type) {
      case 'checkbox':
        value = el.checked ? 2 : 0
        break
      case 'range':
      case 'select-one':
        value = el.value
        break
      case 'button':
      case 'submit':
        value = '1'
        break
      default:
        return
    }

    const query = `${document.location.origin}/control?var=${el.id}&val=${value}`

    fetch(query)
      .then(response => {
        console.log(`request to ${query} finished, status: ${response.status}`)
      })
  }
  // Leer valores inciales y respuestas
  fetch(`${document.location.origin}/status`)
    .then(function (response) {
      return response.json()
    })
    .then(function (state) {
      document
        .querySelectorAll('.default-action')
        .forEach(el => {
          updateValue(el, state[el.id], false)
        })
    })
   // Valor por defecto
  document
    .querySelectorAll('.default-action')
    .forEach(el => { el.onchange = () => updateConfig(el) })


  // Functions for Controls via Keypress
  var keyarriba = 0;
  var keybabajo = 0;
  // Emulate Keypress with Touch
  var arribapress = new KeyboardEvent('keydown', {'keyCode':38, 'which':38});
```

```
  var arribarelease = new KeyboardEvent('keyup', {'keyCode':38, 'which':38});
  var abajopress = new KeyboardEvent('keydown', {'keyCode':40, 'which':40});
  var abajorelease = new KeyboardEvent('keyup', {'keyCode':40, 'which':40});

  //Keypress Events
   document.addEventListener('keydown',function(keyon){
    keyon.preventDefault();
      if ( (keyon.keyCode == '38') && (!keybabajo) && (!keyarriba)) {keyarriba =
1;}
      else if ((keyon.keyCode == '40') && (!keyarriba) && (!keybabajo)){keybabajo
= 1;}
    });

    //KeyRelease Events
    document.addEventListener('keyup',function(keyoff){
      if ((keyoff.keyCode == '38') || (keyoff.keyCode == '40')) {keyarriba =
0;keybabajo = 0;}
    });
    //Send Commands to Scout. Dependiendo de la direccion a tomar, manda a car
una variable u otra
    var currentcommand=0;
    var oldcommand=0;

    window.setInterval(function(){
      if (keyarriba) {currentcommand = 1;}
      else if (keybabajo) {currentcommand = 2;}
      else {currentcommand = 3;}

      if (currentcommand != oldcommand){

fetch(document.location.origin+'/control?var=stepper&val='+currentcommand);
        oldcommand = currentcommand;}

    }, 100);

    </script>
    </body>

</html>
```