



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Esteso

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

Resumen

Este proyecto trata del desarrollo de un robot de tipo excavadora basado en un manipulador móvil. En primer lugar, se analiza el funcionamiento y componentes de una excavadora convencional y el robot meArm. A continuación, y en consecuencia de lo anterior, se selecciona la electrónica para ser programado con Arduino y se realiza el diseño y ensamblaje en el software *SolidWorks*. Después, se realiza una exportación y preparación para la simulación en el software *V-REP CoppeliaSim* y la programación del robot en el software *Arduino IDE*. Finalmente, se realiza el montaje real y se lleva a cabo la escena programada previamente.

Palabras clave: Excavadora, robot, manipulador móvil, SolidWorks, Arduino.

Abstract

This project deals with the development of an excavator-type robot based on a mobile manipulator. Firstly, the operation and components of a conventional excavator and the meArm robot are analysed, followed by the selection of the electronics to be programmed with Arduino and the design and assembly in SolidWorks software. Afterwards, an export and preparation for the simulation in the V-REP CoppeliaSim software and the programming of the robot in the Arduino IDE software are carried out. Finally, the final assembly is performed and the previously programmed scene is carried out.

Key words: Excavator, robot, mobile manipulator, SolidWorks, Arduino.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Documento nº1:
Memoria

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Esteso

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

Contenido

1.	Introducción	5
1.1.	Objetivos	5
1.2.	Justificación y motivación del proyecto	5
2.	Antecedentes.....	6
2.1.	Manipuladores Móviles	6
2.2.	Robot manipulador: el robot meArm	7
2.3.	Plataforma Arduino	9
2.3.1.	Conceptos básicos	9
2.3.2.	Placa Arduino	9
3.	Estudio de necesidades.....	11
3.1.	Máquinas excavadoras.....	11
3.2.	Alcance y limitaciones	12
4.	Planteamiento de soluciones alternativas y justificación de la solución adoptada	12
5.	Metodología.....	13
6.	Análisis máquina excavadora y robot MeArm	14
7.	Selección de la electrónica	15
7.1.	Parte móvil	15
7.2.	Parte de brazo manipulador robótico	16
7.3.	Componentes para el control.....	17
7.4.	Batería.....	18
8.	Diseño y ensamblaje	18
8.1.	Preparación previa.....	19
8.1.1.	Implementación de piezas adquiridas.....	19
8.1.2.	Componentes electrónicos	22
8.1.3.	Robot meArm	23
8.2.	Parte móvil	23
8.2.1.	Laterales.....	24
8.2.2.	Base	26
8.2.3.	Parte frontal.....	28
8.2.4.	Ensamblaje de la parte móvil.....	29
8.3.	Parte de brazo manipulador robótico	30
8.3.1.	Desplazamiento de la estructura.....	30
8.3.2.	Pala	32
8.3.3.	Cabina	33
8.4.	Ensamblaje final	34

9. Exportación para la simulación en V-REP (CoppeliaSim)	35
10. Programación.....	39
10.1. ESP32-CAM.....	39
10.2. Arduino IDE.....	41
11. Bibliografía	52

Tabla de ilustraciones:

Ilustración 1: Concepto del robot "Curiosity" desarrollado por la NASA.....	6
Ilustración 2: Ejemplos de manipuladores móviles en el mercado.....	7
Ilustración 3: Robot paralelogramo FANUC M-410	8
Ilustración 4: Robot meArm de Sistemas Robotizados.....	8
Ilustración 5: Logotipo de Arduino.....	9
Ilustración 6: Software Arduino IDE	10
Ilustración 7: Partes de una excavadora	11
Ilustración 8: Metodología llevada a cabo	13
Ilustración 9: Esquema comparativo máquina excavadora y robot meArm	14
Ilustración 10: Motorreductor amarillo	15
Ilustración 11: Pack de orugas + motor DC + ruedas + pieza de unión, adquirido de AliExpress.....	16
Ilustración 12: Servomotor SG90	16
Ilustración 13: Sensor de ultrasonidos HC-SR04	17
Ilustración 14: Módulo ESP32-CAM.....	17
Ilustración 15: Soporte de pilas 18650	18
Ilustración 16: Perfil seccionado de la pieza de rueda motriz en el proceso de diseño implementada	19
Ilustración 17: Componentes de un engranaje.....	20
Ilustración 18: Rueda motriz implementada en SolidWorks.....	20
Ilustración 19: Rueda conducida y pieza de unión rueda motriz – motor DC implementadas en SolidWorks.....	21
Ilustración 20: Pieza de rueda de oruga implementada en SolidWorks.....	22
Ilustración 21: Esquema de componentes electrónicos y sus usuarios	23
Ilustración 22: Soporte de batería implementado en SolidWorks	23
Ilustración 23: Estructura inicial del robot meArm proporcionada en la asignatura de Sistemas Robotizados	23
Ilustración 24: Estructura de la que se parte para el diseño del brazo manipulador	23
Ilustración 25: Esquema del sistema de rueda de oruga.....	24
Ilustración 26: Dimensiones motor DC.....	25
Ilustración 27: Vista isométrica de la estructura lateral derecha de la parte móvil	25
Ilustración 28: Vista isométrica de la estructura lateral izquierda de la parte móvil	25
Ilustración 29: Soporte de pilas 18650 y sus medidas.....	26
Ilustración 30: Dimensiones servomotor SG90	27
Ilustración 31: Pieza de sujeción de servomotor (nº10 del manipulador móvil)	27
Ilustración 32: Vista isométrica de la base de la parte móvil	28
Ilustración 33: Vista isométrica de la pieza frontal de la parte móvil.....	29
Ilustración 34: Dimensiones HC-SR04.....	29
Ilustración 35: Sistema de rueda de oruga implementado.....	29
Ilustración 36: Perfil seccionado de la parte móvil.....	30
Ilustración 37: Vista isométrica de la parte móvil.....	30
Ilustración 38: Resultado del desplazamiento de la estructura de la parte de manipulador robótico	31
Ilustración 39: Vistas isométricas de las piezas nº2, nº3, nº5 y nº11 respectivamente, de la parte de robot manipulador, afectadas en el desplazamiento de la estructura. ...	31
Ilustración 40: Vistas isométricas de las piezas originales nº2, nº3 y nº5 respectivamente, del robot meArm, afectadas en el desplazamiento de la estructura. ...	31
Ilustración 41: Piezas del extremo del manipulador robótico.....	32
Ilustración 42: Diseño de pala desarrollado	32

Ilustración 43: Dimensiones ESP32-CAM	33
Ilustración 44: Proceso de diseño de la base del manipulador	34
Ilustración 45: Estructura de cabina diseñada.....	34
Ilustración 46: Resultado final del modelado de robot excavadora implementado.....	35
Ilustración 47: Logotipo de Github.....	35
Ilustración 48: Ejemplo de Sistemas de Referencia definidos	36
Ilustración 49: Posición del robot para su programación	36
Ilustración 50: Interfaz y jerarquía establecida en SW2URDF	36
Ilustración 51: Sistemas de coordenadas utilizados	37
Ilustración 52: Ejemplo de uso de SW2URDF	37
Ilustración 53: Sistema de coordenadas de la pala	37
Ilustración 54: Resultado de exportación en CoppeliaSim	38
Ilustración 55: Jerarquía del robot meArm hasta el eje del efector final (pala en este caso).....	39
Ilustración 56: Módulo USB a TTL	40
Ilustración 57: Diagrama de conexiones Arduino UNO y ESP32-CAM.....	40
Ilustración 58: Código ESP32-CAM CameraWebServer	41
Ilustración 59 Ventana de herramientas para programa ESP32-CAM.....	41
Ilustración 60: Interfaz de aplicación ESP32-CAM	41
Ilustración 61: Articulaciones del robot excavadora	42
Ilustración 62: Dimensiones del manipulador del robot excavadora 1 ($q_1=90^\circ$, $q_2 >90^\circ$ y $q_3 <90^\circ$)	43
Ilustración 63: Dimensiones del manipulador del robot excavadora 2 ($q_1=q_2=q_3=90^\circ$)	43
Ilustración 64: Tabla de valores de las dimensiones	44
Ilustración 65: Ubicación pw0 y pT.....	44
Ilustración 66: Esquema de funcionamiento establecido	45
Ilustración 67: Distancia entre el sistema de coordenadas y la lente del sensor HC-SR04	46
Ilustración 68: Distancia entre el eje del servomotor de la pala y extremo de la pieza del manipulador nº22	46
Ilustración 69: Posición inicial del robot Pt0 con pala cerrada	47
Ilustración 70: Posición Pt1 con pala abierta.....	47
Ilustración 71: Posición Pt2 del robot con pala cerrada.....	47
Ilustración 72: Control establecido con mando de IR	48
Ilustración 73: Resultado final del montaje.....	49

1. Introducción

1.1. Objetivos

Este proyecto tiene como objetivo principal realizar un robot de tipo excavadora basado en un manipulador móvil, pasando por los procesos de selección de electrónica, diseño, programación y fabricación. Este tipo de robot se ha basado en la propia máquina real, la cual combina una parte móvil accionada por neumáticos u orugas, junto a una parte fija con estructura de brazo manipulador robótico, con el objetivo de accionar una pala para la realización de las tareas propias de una máquina de este tipo.

En primer lugar, se analiza este tipo de máquina y su funcionamiento para determinar en consecuencia los componentes que serán necesarios en la electrónica con objetivo de simular tanto su funcionamiento como aspectos físicos de su estructura. Tras analizar la electrónica y componentes necesarios, se pasa al proceso de adquisición, estudiando las posibles opciones de compra.

En base a las previas decisiones tomadas, se plantea el resto del diseño de la estructura mediante el programa Solidworks, software CAD para modelado mecánico tanto en 2D como en 3D. Se realiza un diseño completo para la parte móvil relativa a las ruedas y una modificación de la estructura del robot MeArm, desarrollado y versionado en la asignatura Sistemas Robotizados, para la parte de estructura de brazo manipulador robótico. Durante el proceso de diseño, se crea un ensamblaje dentro del mismo programa donde se incorporan todos los componentes electrónicos elegidos y las piezas diseñadas.

El ensamblaje final es preparado y exportado al software de simulación V-REP (CoppeliaSim) donde se plantea una simulación teórica del mismo. Se estudia a su vez la estructura final del robot con objetivo de controlar el efector final, en este caso la pala.

Por último, se desarrolla la programación mediante lenguaje Arduino IDE y se procede al montaje y simulación de una escena real del robot en base a la programación.

Como objetivos adicionales, se tiene la realización del proyecto con materiales asequibles económicamente.

1.2. Justificación y motivación del proyecto

A lo largo del grado, se han estudiado múltiples ramas y ámbitos relacionados con la electrónica industrial y la automática. Para la realización de este trabajo de fin de grado, se buscaba algo que englobase de cierta manera varios aspectos de los vistos a lo largo de estos años. Es aquí donde entran los sistemas robotizados, los cuales combinan varios aspectos relacionados con la electrónica, automática y programación entre otros.

A su vez, resulta interesante el proceso de representar y simular el funcionamiento y modelo de una máquina o un entorno industrial cotidiano, por ejemplo. Es en este punto donde aparece la idea de representar algo tan común como puede ser una máquina excavadora, partiendo de su funcionamiento y las herramientas disponibles para realizar y materializar dicha idea.

Además, se parte de una versión del robot meArm visto en la asignatura de Sistemas Robotizados. La versión original de este robot se encuentra de forma gratuita en distintos repositorios online de modelos CAD. La motivación de ello es crear una variante de este robot que no solo se limite a un brazo manipulador si no que incluya un sistema de movimiento convirtiéndolo en lo que se conoce como un manipulador móvil, los cuales en la actualidad se encuentran en continuo desarrollo.

2. Antecedentes

2.1. Manipuladores Móviles

Actualmente, el termino de manipulador móvil es utilizado para referirse a sistemas robotizados formados por una plataforma móvil y un brazo manipulador robótico. Estos combinan las ventajas de ambos sistemas, permitiendo una gran movilidad además de poder realizar las tareas de manipulación tan comunes en robots manipuladores industriales.

La plataforma móvil ofrece un espacio de trabajo ilimitado para el manipulador, el cual puede contar también con diferentes grados de libertad. Esto ofrece múltiples posibilidades al usuario y al mismo tiempo, supone un gran desafío al tener que tratar con un espacio continuamente cambiante. Es por ello por lo que actualmente es un tema de gran interés en los entornos de desarrollo e investigación.

En función del método de control, los manipuladores móviles pueden ser de dos tipos: autónomos u operados a distancia (“teleoperados”). Tanto unos como otros ya son utilizados en muchas áreas como exploración espacial, atención médica, atención domiciliaria u operaciones militares. En cambio, en el ámbito industrial su implementación ha sido limitada, una razón es el hecho de asumir ciertos riesgos en la implantación de nuevas tecnologías. Además, sus necesidades de desarrollo siguen estando presentes y se cuenta en el mercado con plataformas móviles, sistemas de visión artificial y manipuladores, los cuales integran estos dispositivos.



Ilustración 1: Concepto del robot "Curiosity" desarrollado por la NASA

Un manipulador móvil puede dividirse en plataforma de robot móvil, brazo manipulador robótico y efector final, el cual debe seleccionarse en función del elemento a tratar. Incluyen además diferentes funciones entre las que destaca el procesamiento de imágenes o el control de coordinación, para evitar colisiones entre el conjunto formado por la parte móvil y brazo manipulador. Además, pueden contar con un sistema de seguridad si estos están pensados para trabajar junto a personas.

Sin embargo, estos sistemas también cuentan con varias desventajas. En primer lugar, aún no son lo suficientemente flexibles para ser utilizados en tareas que cambian rápidamente y sus sistemas de control requieren un gran desarrollo, es por ello por lo que no son comúnmente utilizados en pequeñas y medianas empresas. Aún así, su

desarrollo es cada vez mayor y se están estudiando para la realización de tareas tales como servicios de entrega a domicilio, cuidado de mayores o cosecha.

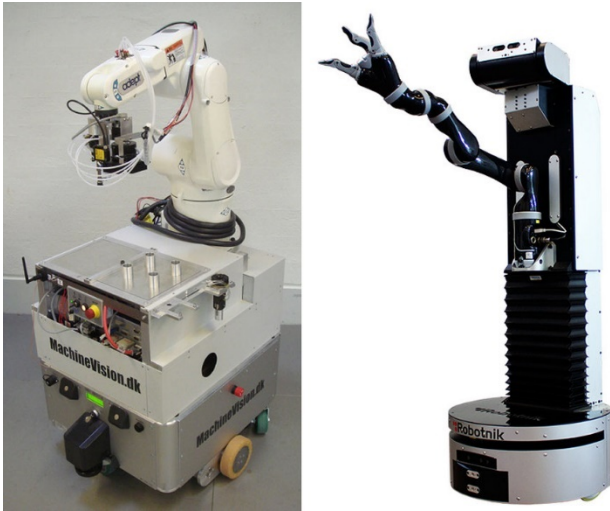


Ilustración 2: Ejemplos de manipuladores móviles en el mercado

2.2. Robot manipulador: el robot meArm

Según la AFNOR ('Association française de Normalisation'), un **robot manipulador** puede definirse como un "mecanismo formado por elementos en serie articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico".

El efector final de un robot manipulador puede contar como máximo con **6 grados de libertad** respecto a sus ejes, 3 asociados a la posición y 3 a la orientación. Puede tener redundancia, es decir, puede tener capacidad para la realización de una misma tarea en distintas configuraciones según el número de articulaciones de la cadena cinemática.

A su vez, la **cadena cinemática** que forman sus articulaciones puede ser en serie (grado de conectividad = 2) abierta o cerrada, o en paralelo (grado de conectividad ≥ 3), definiéndose como grado de conectividad al número de eslabones unidos a un eslabón a través de una articulación.

Atendiendo a estas características, se pueden encontrar diferentes tipos de robots:

- Robot antropomórfico: cuenta con 6 grados de libertad, con cadena cinemática abierta en serie y es el más usado en aplicaciones industriales. Su estructura se asemeja a la de un brazo humano.
- Robot Scara: cuenta con 4 grados de libertad, con cadena cinemática abierta en serie y es utilizado para aplicaciones industriales en inserción de componentes.
- Robot cartesiano: cuenta con 3 grados de libertad, con cadena cinemática abierta en serie, de tipo industrial.
- Robot paralelo: cuenta con cadena cinemática cerrada en paralelo y puede ser de dos tipos: plataforma Stewart o robot delta.

- Brazo robot paralelogramo: cuenta con cadena cinemática cerrada en paralelo y se caracteriza porque el efector final está siempre en horizontal.



Ilustración 3: Robot paralelogramo FANUC M-410

El **robot meArm** es un manipulador robótico de 3 grados de libertad con pinza. Dentro de los diferentes tipos de robots manipuladores, podría clasificarse como un robot de tipo paralelogramo puesto que mantiene la posición del efector final siempre paralela al suelo. Su producción es de bajo coste y su diseño libre, disponible en diferentes webs de repositorios 3D.

Durante la asignatura de Sistemas Robotizados impartida en la UPV por el tutor de este proyecto, se trabajó con una versión de este robot tal y como se muestra en la ilustración 4.

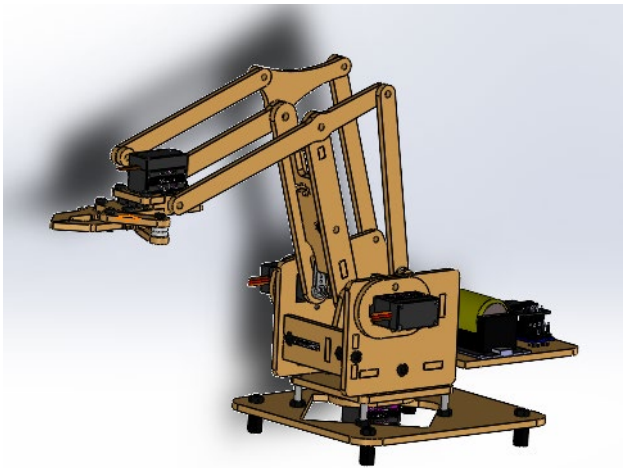


Ilustración 4: Robot meArm de Sistemas Robotizados

Para su realización se tienen distintas opciones en relación con su electrónica: como procesador puede utilizarse un Arduino (Uno, Nano, etc.), una tarjeta ESP32 (ESPduino, ESPDevKit, Wemos D1 R32) o ESP8266 (ESP-01 con programador USB, NodeMCU, Wemos D1 R2); como motores se pueden utilizar servomotores modelos SG90, TS90M o MG90S junto con una controladora (PCA9685, ToroBot); y como baterías una pila de 9V con adaptador, un conjunto de 4 pilas de 1,5 V AA o AAA, una powerbank para pila 16340 o una batería de litio con cargador y

adaptador.

Para su control, se tienen en cuenta sus dimensiones y cálculos adjuntados como anexo (anexo I). Estos tienen en cuenta tanto la cinemática directa del robot como su cinemática inversa.

El **modelo cinemático directo** consiste en la determinación de la posición y orientación del efector final, en este caso una pinza, con respecto a un sistema de referencia

(tomado en la base) y a partir de unos valores de posición de las articulaciones y de los parámetros geométricos del robot.

El **modelado cinemático inverso** consiste en determinar los valores de posición de las articulaciones que son necesarios para que el efector final alcance una posición y orientación deseadas, conocidos también sus parámetros geométricos.

2.3. Plataforma Arduino

2.3.1. Conceptos básicos

Se define como **sistema electrónico** a un conjunto de sensores, actuadores, fuente de alimentación y circuitería o sistemas de control y procesamiento.

- Los sensores se encargan transformar información obtenida en el mundo físico exterior, en una señal eléctrica que puede ser tratada y manipulada por el sistema de control. Los hay de varios tipos, siendo los más típicos de temperatura, movimiento o sonido, entre otros.

- Los circuitos de control son los encargados de procesar la señal eléctrica de forma adecuada, dependiendo de los distintos componentes del sistema.

- Los actuadores son los encargados de transformar la señal eléctrica procesada en energía que actúe en el mundo físico exterior. Ejemplos básicos son una lámpara mediante energía lumínica, un altavoz mediante energía acústica o un motor mediante energía mecánica.

- La fuente de alimentación se encarga de proporcionar al sistema la energía necesaria a todo el sistema para la realización de los distintos procesos. Estas son las baterías, pilas, adaptadores de red, etc.

Un **microcontrolador** es un chip o circuito integrado reprogramable, capaz de ejecutar una serie de instrucciones definidas por el usuario de forma autónoma. Este es el componente principal del circuito de control y procesamiento. Está formado por CPU (Unidad Central de Proceso), diferentes tipos de memorias y pines o patillas de entradas y salidas para la comunicación con el exterior.

2.3.2. Placa Arduino

Arduino es una plataforma de desarrollo basada en placa electrónica. Está formada por tres elementos: una placa electrónica de hardware libre con microcontrolador, un software y un lenguaje de programación.

Una placa de electrónica es una **PCB** (*Printed Circuit Board*), es decir, una placa de circuito impreso. Estas son superficies fabricadas de un material no conductor (resinas de fibra de vidrio reforzado,

cerámica o plástico) en las cuales se laminan o pegan vías de material conductor (cobre). Estas últimas conectan los componentes electrónicos soldados a la placa. En



Ilustración 5: Logotipo de Arduino

comparación a una breadboard u otras del tipo, esta es más estable y compacta. Sin embargo, su diseño no se puede modificar fácilmente una vez se ha fabricado.

El software de Arduino, **Arduino IDE** (ilustración 5), es libre y multiplataforma. Este permite la programación la placa mediante el lenguaje de programación, el cual también es libre y se basa íntegramente en el código C/C++.

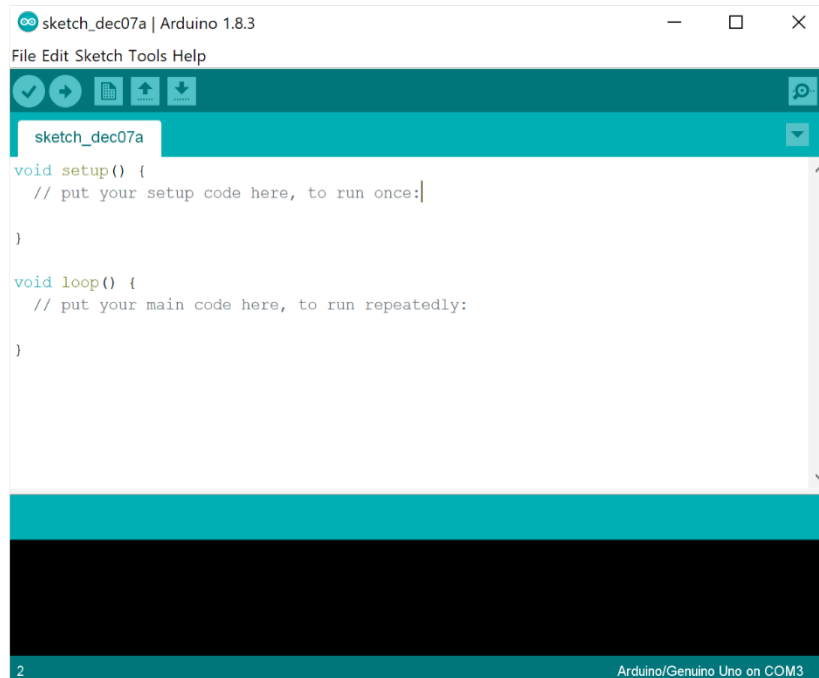


Ilustración 6: Software Arduino IDE

Existen varios modelos de placas Arduino oficiales: UNO, Nano, Mega, Leonardo y Yun, siendo el más común el modelo UNO. Este además es el que se utiliza para este proyecto.

El **microcontrolador** que emplea Arduino UNO es el modelo ATmega328P, con arquitectura de la familia AVR desarrollada por Atmel, de 8 bits y 16 MHz. Este contiene 3 tipos de memorias: memoria Flash (persistente), memoria SRAM (volátil) y memoria EEPROM (persistente).

El voltaje de funcionamiento de Arduino UNO es de 5 V, alimentación que puede conseguirse conectando la placa a una fuente externa (adaptador AC/DC o pila) o conectando la placa a un computador mediante USB.

Arduino UNO dispone de 14 pines hembras de entradas o salidas digitales, numeradas de la 0 a la 13. Estos funcionan a 5 V y proporcionan como máximo 40 mA. Dispone también de 6 entradas analógicas numeradas desde A0 a A5, las cuales también podrían configurarse como pines de salida digital si fuera necesario. Puesto que la placa trabaja con valores digitales, incorpora un circuito conversor analógico/digital para digitalizar los valores analógicos.

3. Estudio de necesidades

Para poder realizar un robot basado en el modelo y funcionamiento de una excavadora y en la estructura de un robot meArm, es necesario conocer cómo funciona y qué partes tiene una máquina excavadora real.

3.1. Máquinas excavadoras

Las excavadoras son comúnmente utilizadas en los sectores de la edificación, construcción y minería. Sus funciones principales son la excavación y transporte de alta resistencia, pudiendo romper suelo, realizar zanjas y triturar el pavimento entre otras. Su uso supone un aumento de eficiencia y productividad, habiendo varios tipos o modelos:

- Excavadora estándar: compuesta de pluma, palo y cubo en la parte delantera. El palo o brazo tiene capacidad para extenderse y bajar al suelo, así como de retraerse hacia la máquina para cargar materiales. La cabina puede girar 360 grados para realizar operaciones mientras el resto de la máquina permanece inmóvil. La máquina funciona mediante dos orugas en paralelo para una mayor tracción en suelo inclinado y una baja presión sobre el suelo, siendo apta para terrenos blandos.

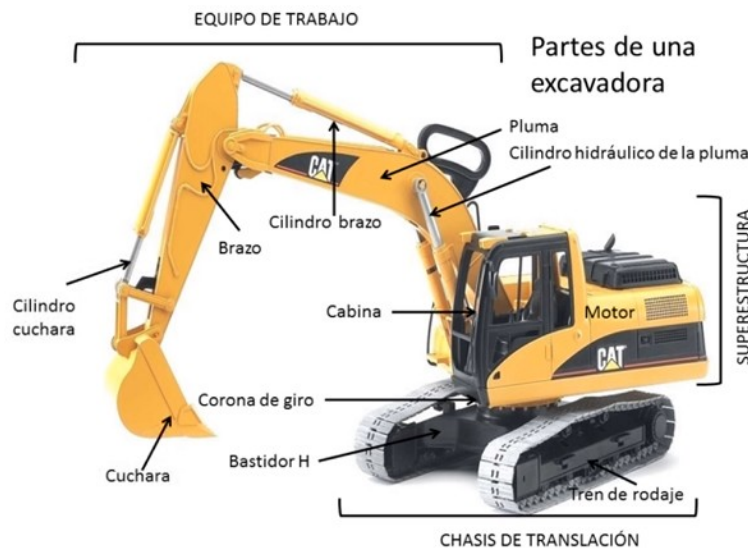


Ilustración 7: Partes de una excavadora

-Excavadora de ruedas: variación de la excavadora estándar que funciona sobre ruedas en lugar de orugas. Estas son más adecuadas para terrenos duros como asfalto y hormigón debido a su menor tracción y mayor presión sobre el suelo.

- Excavadora de largo alcance: variación de la estándar con pluma y brazo extendido. Utilizada para zonas de difícil acceso y tareas de demolición que requieren alta precisión.

- Excavadora de arrastre: también llamadas dragas, son de mayor potencia, utilizadas para aplicaciones a gran escala como en minería y extracción de canales o ingeniería civil.

- Retroexcavadora: presentan el brazo en la parte trasera y en la parte delantera un gran cubo. Su rotación está limitada a 200 grados y son usadas en el paisajismo, construcción y minería.
- Palas hidráulicas: diseñadas para la minería y excavación intensas con cubo de gran tamaño y un brazo adicional de elevación. El cubo puede estar orientado hacia la cabina o hacia fuera.
- Excavadora de succión: es el tipo menos común, permite remover la tierra mediante el uso de agua a presión y un tubo de succión.

3.2. Alcance y limitaciones

El robot de tipo excavadora que se plantea se basa en el modelo estándar y en un manipulador robótico, por lo que se puede dividir en dos partes: una parte móvil y una parte fija de brazo manipulador robótico. El funcionamiento se basa en el movimiento de servomotores y toda la estructura se fabrica en corte láser, por lo que el diseño de las piezas se realiza en dos dimensiones.

La electrónica seleccionada tiene en cuenta el coste de adquisición del material, así como las dimensiones y características de la estructura y componentes del robot a fabricar, relativas a tamaño y peso entre otros. Por esta razón se tienen ciertas limitaciones como el uso de un servomotor SG90 en la base cuyo giro está limitado a 180 grados, mientras que una excavadora estándar permite el giro en 360 grados.

El diseño del modelo se realiza mediante el software **Solidworks** el cual proporciona una gran caracterización de las piezas gracias a sus herramientas de diseño, adecuadas para el modelado en 2D y posterior ensamblaje. Para la estructura de brazo manipulador se parte del robot MeArm cuyo diseño también está realizado en dos dimensiones.

La programación se realiza mediante Arduino IDE, software libre el cual presenta limitaciones de cálculo y memoria frente a otras tarjetas como puede ser Raspberry Pi.

4. Planteamiento de soluciones alternativas y justificación de la solución adoptada

El uso de servomotores más robustos que el SG90 como, por ejemplo, el MG993 o MG90S para la estructura del brazo manipulador a excepción del giro de la base, produciría una mayor precisión en los movimientos y una mejor respuesta. Sin embargo, el tamaño y peso de estos servomotores influiría en el diseño de la estructura de madera de corte láser y en el equilibrio global del robot, además de suponer un mayor coste económico.

El modelado de la estructura en 3D, distinto de corte láser, supondría un diseño mucho más atractivo y robusto. Sin embargo, supondría un mayor coste material, además de no ser estrictamente necesario en un robot de estas características. Con un diseño en 2D pueden alcanzarse los objetivos propuestos además de permitir el uso de electrónica más económica debido a su menor peso como lo anterior mencionado con relación a los servomotores.

En resumen, una mejora en los servomotores implicaría una mejora en el modelado de la estructura y viceversa, lo que agravaría los costes tanto económicos como de producción considerablemente.

Para la parte móvil se plantea el uso de ruedas de oruga propias del modelo estándar, las cuales se adquieren de un comprador de la plataforma de ventas online AliExpress tal y como se trata en el apartado 7. Como alternativa, estas se pueden diseñar en SolidWorks y fabricar mediante impresión 3D, sin embargo, la adquisición directa de este componente y su representación aproximada en el programa resulta más práctica y menos costosa.

5. Metodología

La metodología que se lleva a cabo en este proyecto corresponde con la representada en el organigrama de la Ilustración 8.

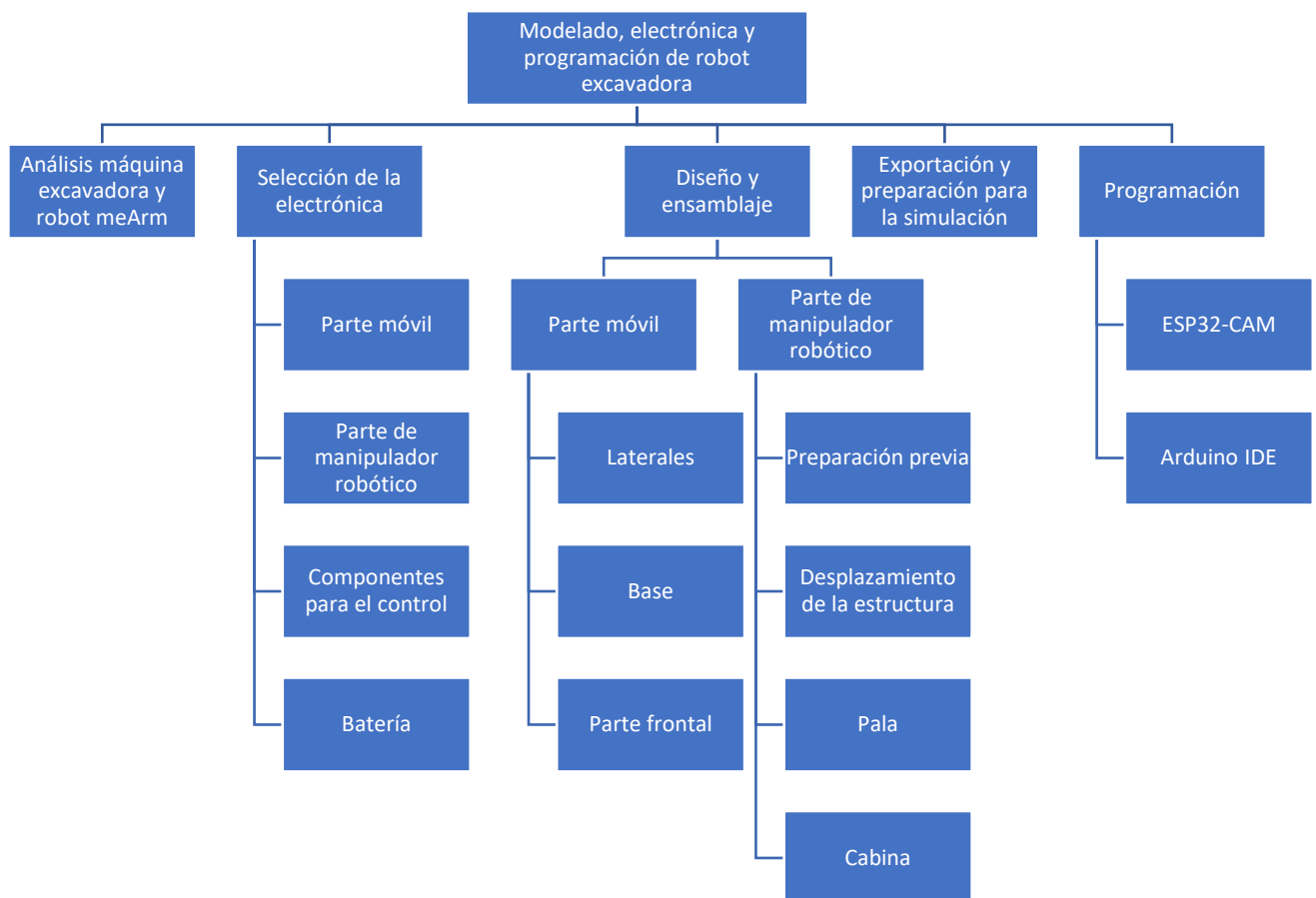


Ilustración 8: Metodología llevada a cabo

En primer lugar, se analiza una máquina excavadora real y el robot meArm puesto que la estructura del manipulador robótico se basa en este. A continuación, se selecciona

la electrónica que se utiliza con objetivo de cumplir las funcionalidades que se plantean. Después, se comienza con el diseño y ensamblaje en el software 3D SolidWorks. Estos se realizan de forma sincronizada, con objetivo de detectar posibles fallos a tiempo y poder corregirlos. Más tarde, se realiza una exportación y preparación para la simulación en el software V-REP CoppeliaSim y, por último, la programación de los componentes y del funcionamiento del robot como conjunto en ArduinoIDE.

6. Análisis máquina excavadora y robot MeArm

Tal y como se ha declarado en el apartado 3.2. Alcance y limitaciones, el diseño se basa en una excavadora estándar de las anteriormente clasificadas en el apartado 3.1. Máquinas excavadoras. Con relación a la parte móvil, se necesita por tanto un sistema que simule el movimiento de las ruedas de oruga tanto mecánicamente como estéticamente.

Con relación a la base de la parte de brazo manipulador, una máquina excavadora estándar permite el giro en 360 grados y tiene el eje de este en la parte central de la estructura de ruedas de oruga. Teniendo esto en cuenta y según lo expuesto en el apartado 3.2. Alcance y limitaciones, se selecciona una electrónica que intente imitar a la máquina real en la mayor medida posible.

El resto de la estructura del brazo de robot manipulador se basa en el robot meArm, por lo que es necesario una comparativa entre estructuras. Esta se ha realizado tal y como se muestra en la ilustración 9. La parte de la pluma en la excavadora se representa con el conjunto de articulaciones verticales que suben desde la base en el robot meArm, la parte del brazo de la excavadora se corresponde con las barras horizontales del robot meArm mientras que, por último, la pala o cuchara de la excavadora corresponde con la pinza en el robot meArm. Para esta última se realiza un diseño completo puesto que se trata de un efector final diferente.

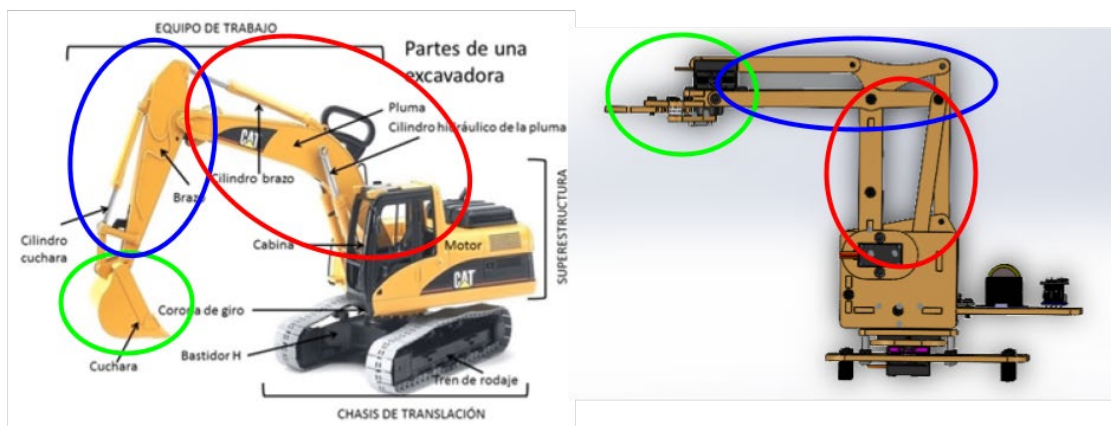


Ilustración 9: Esquema comparativo máquina excavadora y robot meArm

Por tanto y según la estructura del meArm, son necesarios al menos 3 servomotores para el movimiento del brazo, uno de ellos permitirá la rotación de la base y los otros dos permitirán la extensión y ajuste de altura de la parte de la estructura que representa la pluma y el brazo de la excavadora. También se precisa de un cuarto servomotor para el movimiento de la estructura de la pala o cuchara que se diseña.

Por último, una máquina excavadora cuenta con una cabina desde la que se realiza el control o supervisión. Teniendo en cuenta lo estudiado respecto a manipuladores móviles en el apartado 2.1., en el modelo a desarrollar se añade un equivalente a esta parte junto con un sistema de cámara.

7. Selección de la electrónica

Para la selección de la electrónica, se tiene en cuenta en primer lugar que se utiliza una placa **Arduino UNO** para el control del robot como conjunto, por lo que todos los componentes electrónicos seleccionados son compatibles con este. La tensión a la que se alimentan los componentes debe estar entre los 3,3 V y los 5 V, mientras que la corriente necesaria viene determinada por los componentes que se eligen y la alimentación en consecuencia de esta.

7.1. Parte móvil

Se comienza por desarrollar el sistema de la parte móvil que representa el sistema de rueda de oruga de la máquina real, puesto que influirá en la elección del resto de componentes y estructura. Para accionar dicho sistema, se decide que lo más adecuado y económico es un **motor DC amarillo con motorreductor**, comúnmente usado en aplicaciones de robótica y compatible con Arduino. Un motorreductor es un elemento electromecánico el cual permite reducir la velocidad de giro del motor eléctrico a través de un juego de engranajes. Este funciona acoplado al mismo eje del motor eléctrico y se basa en la fórmula del par motor: P (*Potencia*) = $M(\text{Par}) \times \omega$ (*velocidad angular*), de manera

que se consigue dicha disminución de velocidad aumentando el par considerablemente. Para su funcionamiento, requiere un voltaje de alimentación entre 3 V y 9 V, consumiendo una corriente entre 100 mA y 150 mA .



Ilustración 10: Motorreductor amarillo

Para controlar la velocidad y giro de estos motores DC se utiliza un componente **L293D**. Se trata de un circuito integrado utilizado tanto para motores DC como bipolares de paso, puede conducir hasta 600 mA y trabaja con tensiones comprendidas entre 4.5 V y 36 V. Su estructura se basa en configuración de puente H para poder invertir el sentido de giro de los motores.

Puesto que este circuito integrado L293D dispone de pines, los cuales deben tener conexión con otros pines de este y de Arduino, se plantean inicialmente dos opciones. La primera de ellas es adquirir

un módulo de tarjeta comercial que integre este cableado en una PCB (*Printed Circuit Board*). Las ventajas de esto sería un ahorro considerable de cables y las desventajas, que ocuparía mayor espacio en el posterior diseño de la estructura. Por otro lado, la segunda opción es adquirir una **placa de extensión Protoshield**, la cual incorpora una *bread board* y que ensambla con Arduino UNO. La ventaja de usar esta, es que no solo se podrá conectar un componente L293D, sino que también se podrán realizar otras conexiones que necesiten otros componentes. Como desventaja se tendría el uso de numerosos cables. Teniendo en cuenta todo lo anterior, se elige esta segunda opción.

Una vez seleccionado estos componentes, se plantea un sistema de movimiento con dos motores DC con motorreductor a cada lado, de forma que cada motor sea capaz de accionar una rueda motriz, la cual a su vez haga girar una rueda conducida mediante fricción o engranaje de una correa de plástico o de goma. Tras analizar varias opciones disponibles, se decide adquirir un pack de estos tres elementos ofertados por el vendedor "Estone super market" de AliExpress bajo el nombre "Correa transportadora de pista de tanque inteligente TT Motor Drive juegos de ruedas DIY Accesorios". Dicho pack funciona como conjunto por lo que se asegura en una primera instancia que no se tienen problemas de compatibilidad entre piezas.



Ilustración 11: Pack de orugas + motor DC + ruedas + pieza de unión, adquirido de AliExpress

7.2. Parte de brazo manipulador robótico

Respecto a la parte de estructura fija de brazo manipulador robótico y según lo detallado en el apartado 6. Análisis máquina excavadora y robot meArm y 3.2. Alcance y limitaciones, todos los **servomotores** que se seleccionan inicialmente son del modelo



Ilustración 12: Servomotor SG90

SG90. Este es un servo miniatura de bajo consumo, de unos 20-30 mA alimentado entre 4,8 V y 6 V, pequeñas dimensiones y muy económico, comúnmente utilizado en aplicaciones de electrónica y robótica. Para el caso de la base que permite el giro, este se limita a 180 grados frente a los 360 grados de la máquina real. En caso de tener problemas de estabilidad debidos al peso y organización de la estructura, se plantearía el cambio específico de este servomotor por uno del

modelo MG90S, de mayor robustez y similares dimensiones. Los otros tres servomotores incluido el que mueve la pala en el diseño que se desarrolla, mueven un peso similar a como lo hacen en el robot meArm por lo que no se espera tener problemas en ese aspecto.

7.3. Componentes para el control

Con fines de control, se añade al frente de la estructura móvil un **sensor de ultrasonidos HC-SR04**. Este es un sensor para medir distancia en un rango entre 2 cm y 450 cm, de bajo coste, bajo consumo de unos 15 mA en uso y 2 mA en reposo y buena precisión. Es por ello por lo que es el más utilizado en proyectos de robótica. Dispone de dos transductores: un emisor y un receptor piezoeléctricos. Funciona de manera que el emisor piezoeléctrico emite 8 pulsos de ultrasonido a 40 kHz tras recibir la señal de activación en el pin TRIG. Tras ello, las ondas viajan en el aire hasta rebotar en algún objeto. Esta señal rebotada es recogida por el pin ECHO y según el tiempo transcurrido desde el comienzo de este bucle, se calcula la distancia mediante la fórmula:



Ilustración 13: Sensor de ultrasonidos HC-SR04

$$\text{Distancia (m)} = (\text{Tiempo de pulso (seg)} \times \text{velocidad del sonido (m/s)})/2$$

Siendo la velocidad del sonido = 343 m/s = 0.0343 cm/ μ s y la fórmula dividida entre 2 puesto que se mide el tiempo tanto de ida como de vuelta.

Para el control general de ambas partes, móvil y de manipulador robótico, se decide utilizar un **receptor con sensor de infrarrojos**. Este utiliza un puerto de salida digital y junto con un mando de IR, permite el control a una determinada distancia. Puesto que previamente se ha seleccionado la placa *Protoshield* con *bread board*, se puede realizar en esta su conexión.

Para el sistema de control o supervisión comentado en el apartado anterior, a todo el conjunto se le añade un componente externo al Arduino, un **módulo ESP32-CAM**. Se trata de un pequeño módulo de cámara con un chip ESP32-S. Este es un microcontrolador integrado de los denominados “SoC (*System on a Chip*)”, del fabricante *Expressif Systems*. Incluye CPU, memoria interna de 512 kB, consumo de 180 mA con flash apagado alcanzando 310 mA con este encendido, conectividad Wi-Fi y Bluetooth, entre otros. La funcionalidad de este módulo será poder tener una imagen en todo momento de lo que ve la estructura de cabina que se diseña.



Ilustración 14: Módulo ESP32-CAM

7.4. Batería

Por último, con todos los componentes seleccionados se procede a la elección de la batería. Para ello se hace una suma de todas las corrientes puesto que se conectan en paralelo, asumiendo el caso peor de cada componente, es decir, el máximo consumo posible y que todos ellos requieran esa cantidad en el mismo instante. Por tanto:

$$\begin{aligned} \text{Corriente instantánea máxima (mA)} \\ &= 30 \text{ mA} \times 4 (\text{Servos SG90}) + 150 \text{ mA} \times 2 (\text{motores DC}) \\ &+ 310 \text{ mA} (\text{ESP32 - CAM}) + 15 \text{ mA} (\text{HCSR - 04}) = 745 \text{ mA} \end{aligned}$$

Esta corriente es teórica, ofrecida por los fabricantes por lo que diferirá de la real. Sin embargo, será orientativa a la hora de escoger una batería. También se tiene en cuenta que no se sabe en medidas de tiempo cuánto consumirá cada elemento a menos que se midiera la capacidad de una pila en un determinado tiempo con todo el conjunto en funcionamiento, método que en principio se descarta. En consecuencia, se tiene en cuenta otros factores como la experiencia previa con el robot meArm, de 4 servomotores SG90 y pila de 4800 mAh teóricos, la cual había que cargar con considerable frecuencia. Por todo ello y sin descartar inicialmente posibles adiciones de otros componentes o funcionalidades, se decide adquirir **4 pilas 18650** de 9900 mAh, junto con un **soporte**. El vendedor del soporte es “diymore Alice1101983 Store” en la web de AliExpress e indica en este que puede proporcionar hasta 3 A en teoría, por lo que es adecuado. Se tiene en cuenta que la capacidad real proporcionada por estas pilas suele ser bastante inferior a lo indicado.

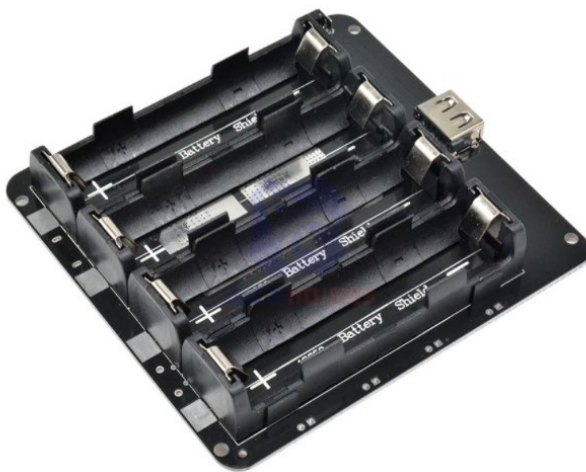


Ilustración 15: Soporte de pilas 18650

8. Diseño y ensamblaje

Para comenzar el diseño, es importante prestar atención a las dimensiones de todos los componentes adquiridos. En primer lugar, se comienza diseñando la parte móvil que ensamblará todos los componentes inferiores puesto que de esta parte depende el resto de la estructura.

El diseño se realiza en el **software 3D SolidWorks**, mediante piezas de encaje que tendrán un grosor de 3 mm obtenidas mediante corte láser, misma técnica mediante la cual está diseñado el robot meArm.

Se toman varias consideraciones a la hora de diseñar, una es el uso de tolerancias a lo largo del diseño. Estas se toman de algunos milímetros y no en todos los casos, según en qué situaciones. Por ejemplo, en encajes de componentes con la estructura y tornillos se tiene en cuenta mientras que en la mayoría de encajes entre las propias piezas de la estructura no. Esto facilita el diseño y no supone problemas puesto que a la hora del montaje real se dispone de papel de lija, aprovechando a su vez que el material de fabricación es madera.

Durante el proceso de ensamblaje, se suprime el uso de tornillos puesto que su función en este es estética. Sin embargo, se detalla en el apartado 11 de montaje qué tipo de tornillo requiere cada parte de la estructura. Para las distintas relaciones entre componentes electrónicos y piezas se utiliza la herramienta “relaciones de posición” de SolidWorks.

Tal y como se ha comentado en el apartado 5 de metodología, el motivo de combinar diseño y ensamblaje es comprobar el correcto encaje de todas las piezas y componentes a lo largo del proceso y corregir errores de forma temprana.

8.1. Preparación previa

Antes de comenzar con el proceso de diseño se deben tener, tanto piezas adquiridas como componentes electrónicos, representados e implementados en SolidWorks.

8.1.1. Implementación de piezas adquiridas

Para el caso de estructuras no electrónicas adquiridas, se realiza un modelo en SolidWorks de todas las piezas, de forma que imiten a las originales. En nuestro robot, es el caso del sistema de ruedas de oruga adquirido, formado por el conjunto de rueda motriz, rueda conducida, pieza de unión y oruga.

Se comienza por la implementación de la **rueda motriz**. Según los datos del fabricante, esta tiene un diámetro de 46 mm, aunque al medirla con un calibre se obtienen unos 50 mm, dato que se escoge. El resto de las dimensiones medidas e implementadas corresponden con las de la ilustración 16. Con estos datos, se realiza el croquis inicial mostrado en dicha ilustración a partir del cual, mediante la operación de revolución, se obtiene un sólido con forma de rueda similar al original.

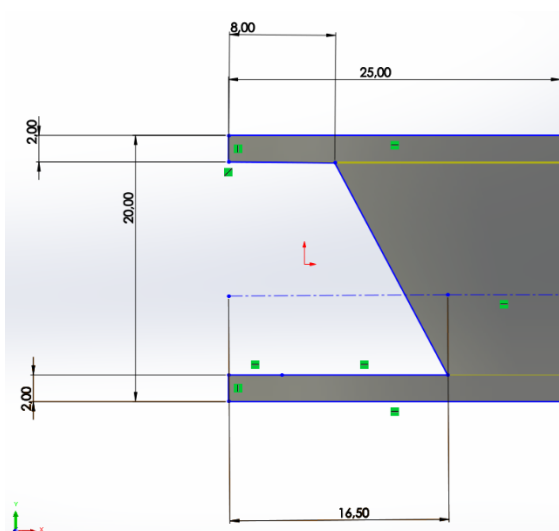


Ilustración 16: Perfil seccionado de la pieza de rueda motriz en el proceso de diseño implementada

Para la representación de los dientes, al ser una pieza de medidas milimétricas, su representación exacta no es posible. Por

ello, esta parte se ha diseñado siguiendo el diseño normalizado de una rueda dentada de engranaje. Para este, se parte de los siguientes datos:

Diametro circunferencia primitiva (d) = 46 mm

Diámetro circunferencia de tolerancia o interior (Ci) = 42mm → Radio (Ri) = 21 mm

Diámetro circunferencia exterior = 50 mm

Nº de dientes (z) = 17

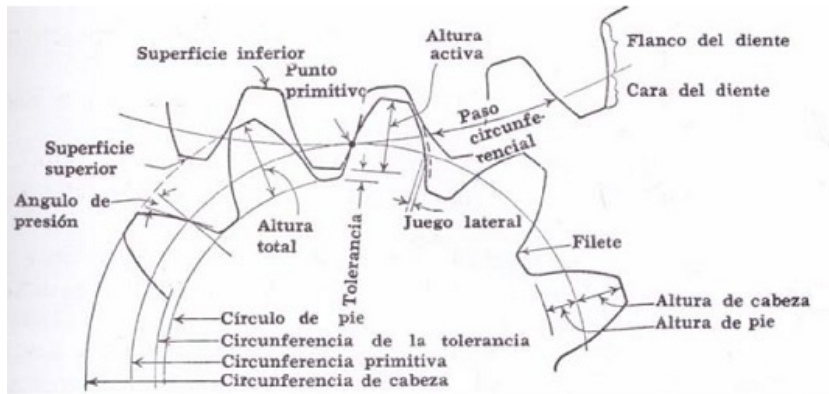


Ilustración 17: Componentes de un engranaje

A partir de los cuales se obtiene:

Modulo(m) = d/z = 2.7059 ≈ 2.75(normalizado)

Juego en cabeza o tolerancia(c) = 0.25 × 2.75 = 0.6875mm

Radio circunferencia base o de tolerancia (Rb) = Ri + c = 21.6875 mm

Cálculo ángulo de presión (α) → Rb = Rp × cos(α) → α = acos(Rb/Rp) = 19.44°

Con el cálculo del ángulo de presión, ya se puede realizar el diseño del diente. Posteriormente se realiza una operación 'matriz' en SolidWorks de 17 dientes, de forma que el resultado es el de la primera imagen de la ilustración 18. Por último, se añade el encaje que presenta esta rueda con la pieza de unión de esta rueda con el motor DC, y todos los detalles de la rueda original, estos últimos con fines estéticos.

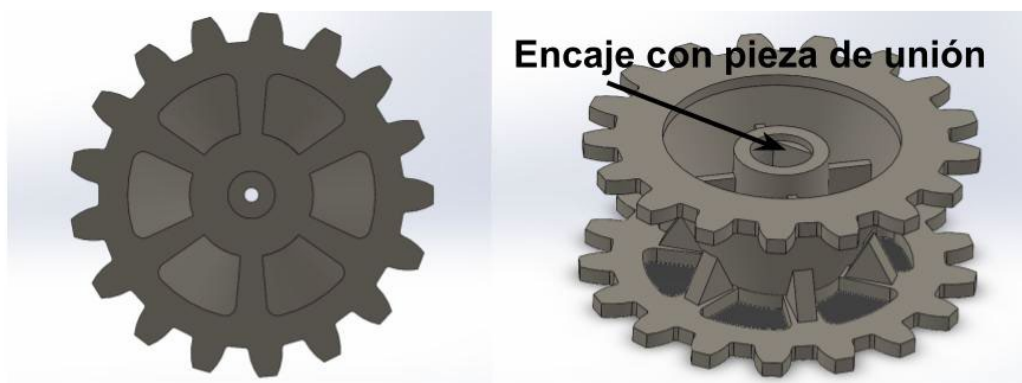


Ilustración 18: Rueda motriz implementada en SolidWorks

Para la representación de la **rueda conducida**, se mide con el calibre, se realiza un croquis y una operación de revolución de igual forma que la rueda motriz, acabando en ese punto. Finalmente, se le añaden detalles estéticos. Para la pieza de unión, se mide

usando de nuevo el calibre y se realizan operaciones básicas de SolidWorks puesto que es sencilla. Por último, se les da color a ambas piezas. El resultado es el de la ilustración 19.

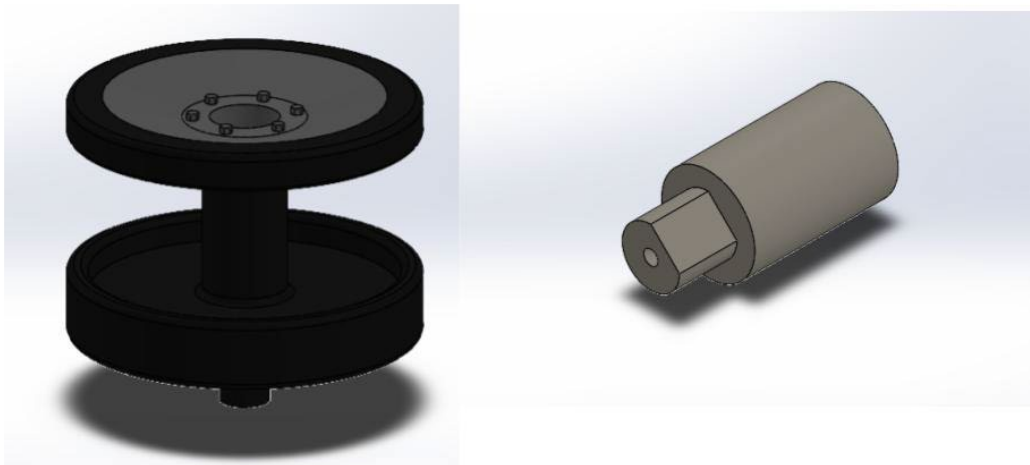


Ilustración 19: Rueda conducida y pieza de unión rueda motriz – motor DC implementadas en SolidWorks

Para representar la rueda de oruga, se observa que está formada de pequeñas piezas enlazadas. Por ello, se representa una de estas piezas y se realiza más adelante una matriz en el ensamblaje de la parte móvil (8.2.4) para representar la cadena.

El conjunto de estas piezas debe engranar con la rueda motriz a través de los agujeros de esta. Por ello, teniendo en cuenta las dimensiones de la rueda motriz y las de esta pieza, se realiza el siguiente cálculo:

$$\text{Longitud interna rueda motriz} = 2\pi \times Ri = 2\pi \times 21 = 131.95 \text{ mm}$$

$$\text{Arco entre centros de piezas} = \frac{131.95}{z \text{ (n}^\circ\text{dientes)}} = \frac{131.95}{17} = 7.762 \text{ mm}$$

Fórmula de la cuerda de una circunferencia:

$$\text{Distancia entre centros de piezas(cuerda)} = 2 \times 21 \times \text{sen}\left(\frac{360/17}{2}\right) = 7.717 \text{ mm}$$

Con esta distancia y teniendo en cuenta que cada pieza con la siguiente en la oruga comparte aproximadamente 2mm, se realiza el diseño de la pieza mediante operaciones básicas de suma y resta. Añadiendo detalles estéticos, se termina obteniendo el resultado de la ilustración 20.

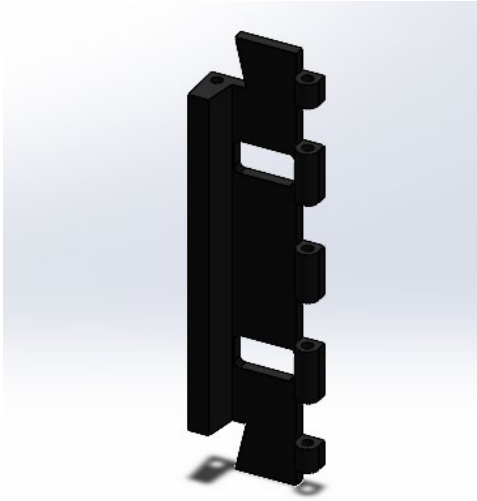


Ilustración 20: Pieza de rueda de oruga implementada en SolidWorks

8.1.2. Componentes electrónicos

Para el caso de los componentes electrónicos, se ha decidido obtener sus modelos de la página <https://www.3dcontentcentral.com/>. Esta se trata de una biblioteca o portal oficial de SolidWorks, donde todos los usuarios tienen la opción de compartir sus modelos creados y ponerlos a disposición del resto de usuarios de la plataforma.

De los componentes adquiridos, se implementan mediante este método los motores DC, servomotores SG90, sensor de ultrasonidos HC-SR04, Arduino UNO, ESP32-CAM y soporte de una pila 18650, de los usuarios del esquema de la ilustración 21. Para el soporte de todas las pilas, se crea una pieza y se realiza un ensamblaje usando este soporte 4 veces, una por cada pila. El resultado se observa en la ilustración 22. En el caso del Arduino UNO, al tener muchos componentes independientes que sobrecargan el programa, se crea una pieza con los componentes más representativos (PCB, microcontrolador, etc.).

Pieza	Usuario de 3DContentCentral
Motor DC amarillo	Manveen Singh
Servomotor SG90	Renato Alonso Hurtado Medina
Sensor ultrasonidos HC-SR04	Benjamin Levesque
ESP32-CAM	Lucas Arthur Baumann
Arduino UNO	Renato Alonso Hurtado Medina
Soporte pila 18650	Yaroslav Tovstopjat

Ilustración 21: Esquema de componentes electrónicos y sus usuarios

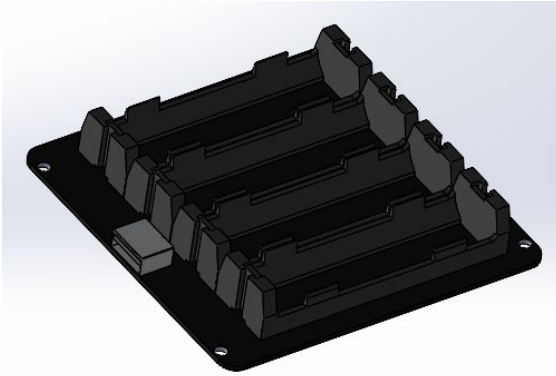


Ilustración 22: Soporte de batería implementado en SolidWorks

8.1.3. Robot meArm

Para la parte del manipulador robótico es necesario preparar la estructura a partir de la del robot meArm de la ilustración 23. Por ello, a partir de la estructura inicial estudiada en el apartado 2.2, se quitan las piezas que no servirán para el diseño. Estas corresponden a las dos piezas de la base y de la pinza. La pieza de la base en la que se encuentra el servomotor que permite la rotación, se sustituye por la pieza de la base de la parte móvil desarrollada en el apartado 8.2.2. De esta forma, la estructura de la que se parte para la parte del manipulador es la que se muestra en la ilustración 24.



Ilustración 23: Estructura inicial del robot meArm proporcionada en la asignatura de Sistemas Robotizados

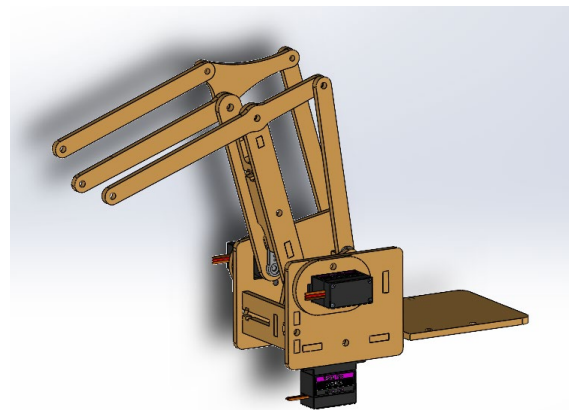


Ilustración 24: Estructura de la que se parte para el diseño del brazo manipulador

8.2. Parte móvil

Se comienza por diseñar la estructura de la parte móvil. Esta debe ensamblar el conjunto de motores DC, ruedas de oruga (rueda motriz, conducida y correa en cada lado), soporte de las pilas, sensor de ultrasonidos HC-SR04 y servomotor de giro de la base de la estructura fija del brazo manipulador, el cual dará pie a dicha estructura en la parte superior. Dicha estructura de la parte móvil se plantea lo más simple y comprimida posible, capaz de albergar todos los componentes anteriores. En consecuencia y tras

varios análisis, se decide realizar esta estructura en tres partes: dos laterales, una base y una parte frontal.

8.2.1. Laterales

Los laterales de la parte móvil ensamblarán los sistemas de rueda de oruga con todos sus componentes junto con los motores DC. Para ello, se necesita saber todas las dimensiones y distancias, por lo que se calcula en primer lugar la distancia entre los ejes de la rueda motriz y la conducida.

Tal y como se ha detallado en el apartado 8.1.1., la rueda motriz tiene un diámetro externo de 50 mm y la rueda conducida un diámetro de 50 mm. El vendedor especifica que el sistema montado en conjunto con la oruga tiene una longitud de 165 mm, dato aproximadamente acertado. Con estas medidas y despreciando la oruga por su complejidad en la realidad, puede calcularse cuánto hay de distancia orientativamente entre los centros de las ruedas:

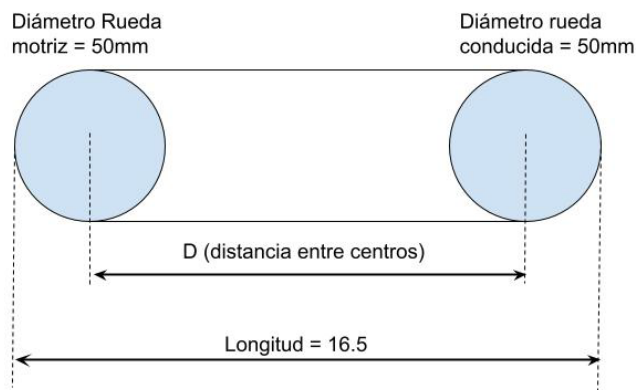


Ilustración 25: Esquema del sistema de rueda de oruga

$$D \text{ (distancia orientativa entre ejes despreciando la oruga)} = 165 - \frac{50}{2} - \frac{50}{2} \\ = 115 \text{ mm}$$

Tras medir esta distancia en el ensamblaje real con ayuda de un calibre y teniendo en cuenta la oruga, se obtiene una medida de distancia entre ejes entre 106 y 107 mm, por lo que se utilizan estas medidas reales para el diseño final. Además, al añadir la rueda de oruga, el eje de la rueda conducida queda ligeramente por encima del de la rueda motriz, por lo que se tiene esto en cuenta en el diseño con objetivo de mantener la base del robot paralela al suelo.

También se debe tener en cuenta la pieza de unión entre el eje del motor y la rueda motriz. Esta pieza tiene un diámetro de 13 mm aproximadamente, medido mediante calibre. Este diámetro es el que se tiene que tomar a la hora de representar el eje de esta rueda en la estructura.

La otra medida para tener en cuenta en este apartado es la de los motores DC, la cual no la proporciona el vendedor, pero se puede encontrar fácilmente en la web de otros vendedores, en este caso el vendedor "Electronilab". Además, se comprueba su veracidad y precisión midiendo el componente real con el calibre y se llega a la conclusión de que las medidas seleccionadas para utilizar son suficientemente acertadas.

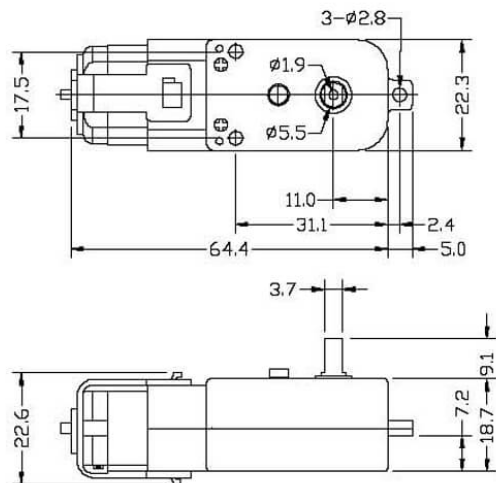


Ilustración 26: Dimensiones motor DC

Estos motores DC tienen en su cuerpo dos pequeños agujeros de diámetro 2.8 mm los cuales sirven para poder ser atornillados a la estructura, método que se utiliza teniendo en cuenta el resto de sus dimensiones en la ilustración 26.

Cada uno de los laterales deberá estar unido a la base de la estructura móvil (8.1.2.) y a la parte frontal (8.1.3.), por lo que se crean unos huecos de encaje para ello.

Para el alto total de la pieza, se tiene en cuenta que ambos motores DC, cada uno a un lado, compartirán espacio con el soporte de las pilas dentro de la cámara interna resultante de la estructura.

Por último, la longitud total de esta pieza vendrá dada por la distribución de los componentes de la base (8.1.2.) y el espesor será de 3 mm debido a la fabricación mediante corte láser.

Con todas estas consideraciones, se realiza el diseño del lateral de la estructura, resultando dos versiones distintas para cada lateral. La diferencia entre ambos es que el lateral izquierdo presenta un agujero para poder apagar el soporte de las pilas.

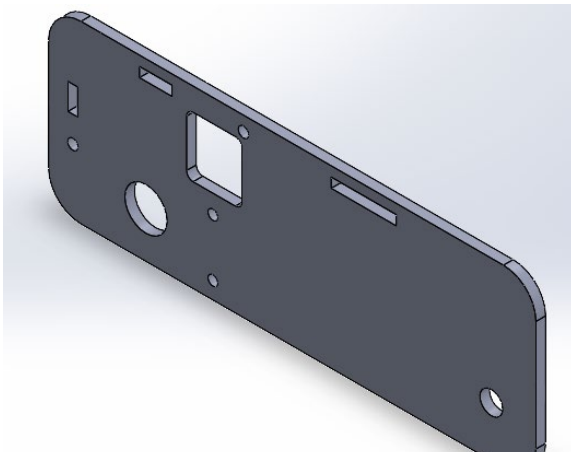


Ilustración 28: Vista isométrica de la estructura lateral izquierda de la parte móvil

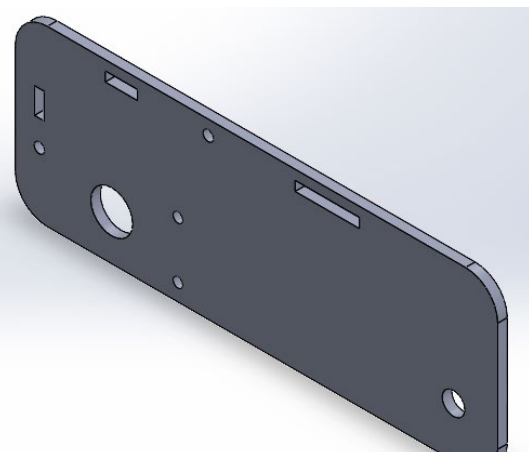


Ilustración 27: Vista isométrica de la estructura lateral derecha de la parte móvil

8.2.2. Base

La estructura de la base tiene como objetivo el ensamblaje del soporte de las pilas, el servomotor que permite el giro de la parte fija de brazo manipulador robótico, los laterales diseñados en el apartado anterior y la parte frontal desarrollada en el siguiente apartado.

Las medidas de los componentes de este bloque son, o bien proporcionadas por el fabricante como en el caso del soporte de las pilas, o bien se pueden encontrar fácilmente en la web de varios fabricantes como es el caso del servomotor SG90. El soporte de las pilas tiene las medidas que se muestran en la ilustración 29.

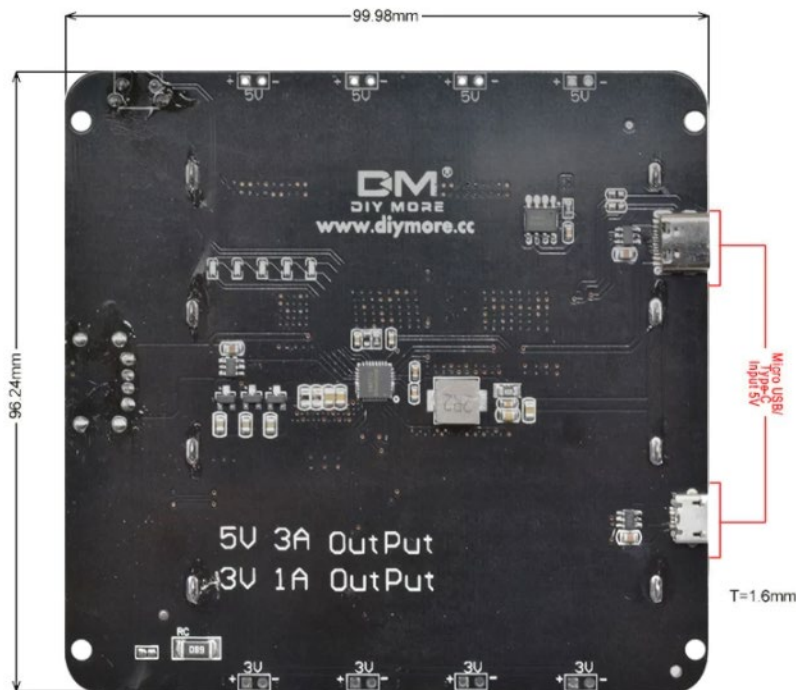


Ilustración 29: Soporte de pilas 18650 y sus medidas

Para el diseño, se tiene en cuenta el puerto de carga de estas pilas a través del soporte para que sea accesible. Estos son los puertos marcados en rojo en la ilustración 29, que se dejarán hacia la parte de atrás del robot. Por ello, el ancho de la base debe tener al menos 96.24 mm.

Respecto al largo, se tiene en cuenta además las dimensiones del servomotor SG90 mostradas en la ilustración 30, obtenidas del vendedor “Alibaba” y comprobadas físicamente con el calibre. En el posicionamiento del servomotor influyen las dimensiones de las piezas de la parte del manipulador robótico (nº1, nº2, nº3 y nº6 desarrolladas en el apartado 8.3) puesto que debe quedar centrado en el

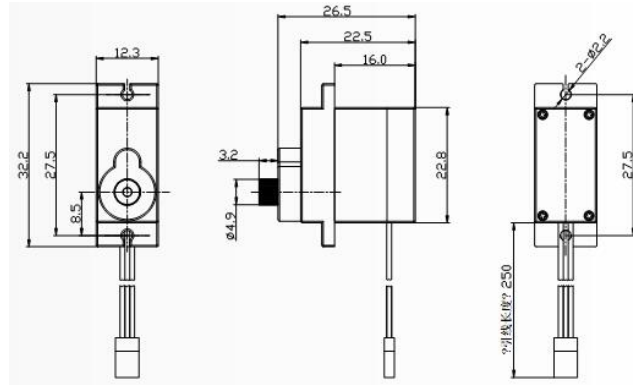


Ilustración 30: Dimensiones servomotor SG90

diseño final. Ambas partes (móvil y manipulador) deben quedar centradas tanto por delante como por los lados, desplazando el eje de giro de dicho servomotor de la base.

Por último, se debe tener en cuenta que hay componentes que quedan debajo de la estructura cuya conexión por cable es necesaria. Por ello se realizan ranuras a una altura aproximada en la que se coloca el Arduino posteriormente en otra pieza (pieza nº1 del manipulador móvil).

Con todas estas consideraciones, se realizó un primer diseño en el que se situó el soporte de la batería en la parte superior de la estructura. Sin embargo, esta chocaría con la estructura del manipulador al girar, por lo que se cambió a la parte inferior en la que se comprobó que no colisionase con el servomotor gracias a las diferencias de altura entre componentes. Por tanto, el soporte se atornilla a la parte inferior con 4 tornillos pasantes M3x25 y tuercas, teniendo en cuenta sus dimensiones en la ilustración 29. Esto influye en el alto del lateral tal y como se detalló en el apartado 8.2.1.

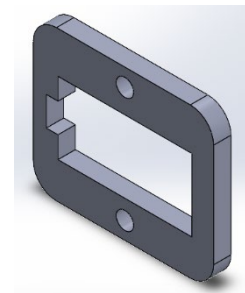


Ilustración 31: Pieza de sujeción de servomotor (nº10 del manipulador móvil)

Para fijar el servomotor a la estructura, se utiliza una pequeña pieza adicional (ilustración 31, pieza nº10 del manipulador móvil).

Finalmente, se obtiene el diseño definitivo de la ilustración 32.

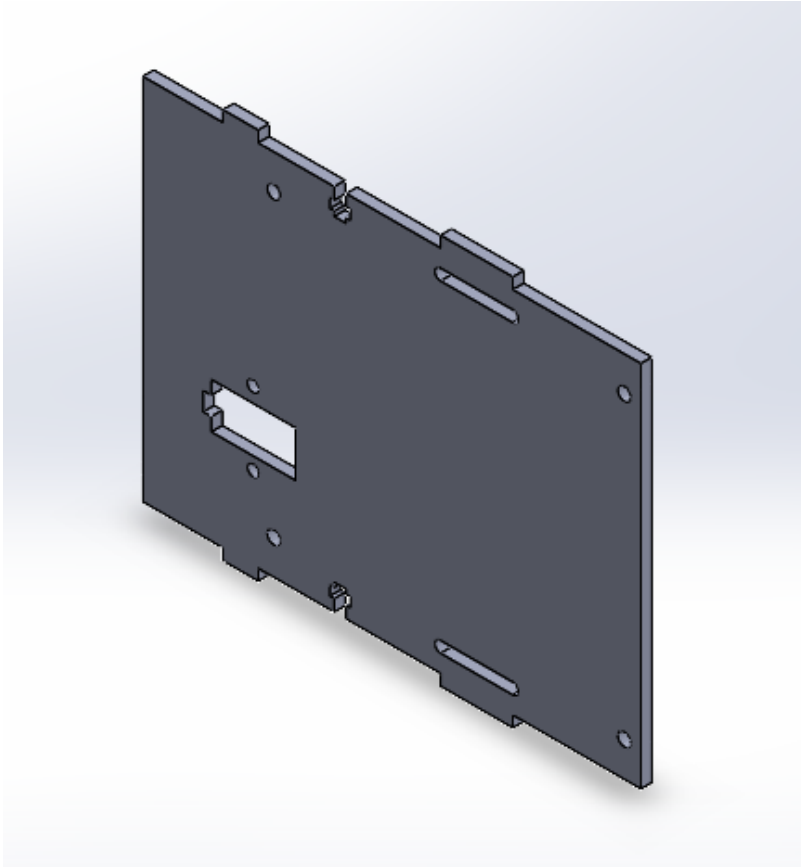


Ilustración 32: Vista isométrica de la base de la parte móvil

8.2.3. Parte frontal

La función de esta pieza en la estructura será proporcionar mayor estabilidad al conjunto de la parte móvil y ensamblar el sensor de ultrasonidos HC-SR04 a la estructura.

Para su diseño, se tienen en cuenta las dimensiones de dicho sensor, las cuales se obtienen de la página “makerguides” y son las de la ilustración 33.

La pieza diseñada tiene el mismo ancho que la base (4.1.2.) y los correspondientes encajes acordes a los de los laterales (4.1.1.). El resultado final es el de la ilustración 34.

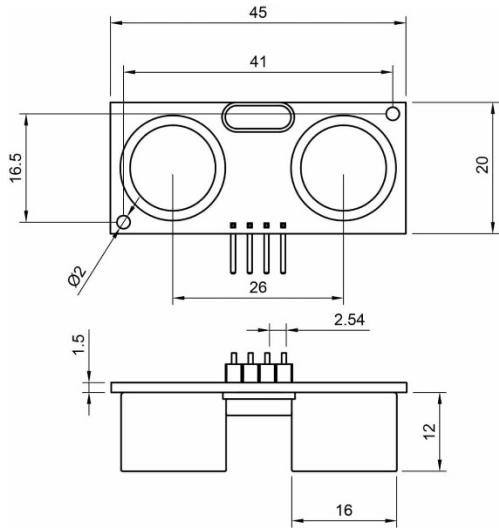


Ilustración 34: Dimensiones HC-SR04

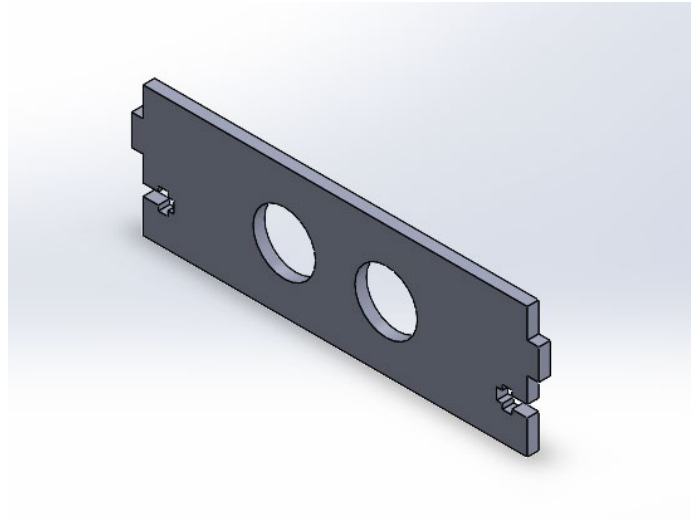


Ilustración 33: Vista isométrica de la pieza frontal de la parte móvil

8.2.4. Ensamblaje de la parte móvil

Se comprueba mediante el ensamblaje el correcto diseño de las piezas. En caso de encontrar desajustes, se corrigen en las piezas.

Para la representación de la rueda de oruga, se realiza una operación matriz de cadena a lo largo de un croquis que define su recorrido. A esta matriz se le asocia la pieza creada en el apartado 8.1.1. junto con las relaciones de enganche de una pieza con la siguiente, obteniendo así el resultado de la ilustración 35.

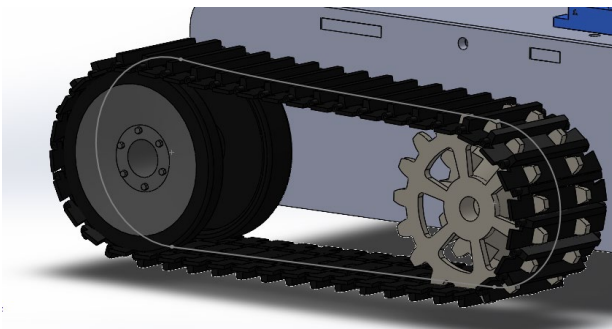


Ilustración 35: Sistema de rueda de oruga implementado

El ensamblaje final de esta parte se muestra en las ilustraciones 36 y 37. En la ilustración 36 se observan el servomotor SG90 y el soporte de la batería superpuestos en la zona marcada en verde, sin embargo, no colisionan ya que están a distinta altura respecto al plano de la pantalla, tal y como se ve en la ilustración 37.

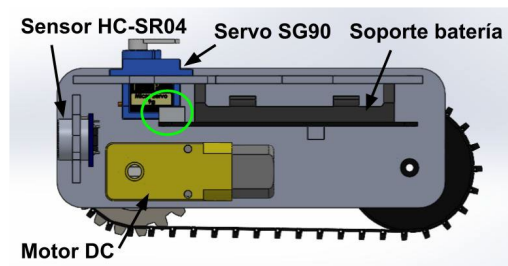


Ilustración 36: Perfil seccionado de la parte móvil

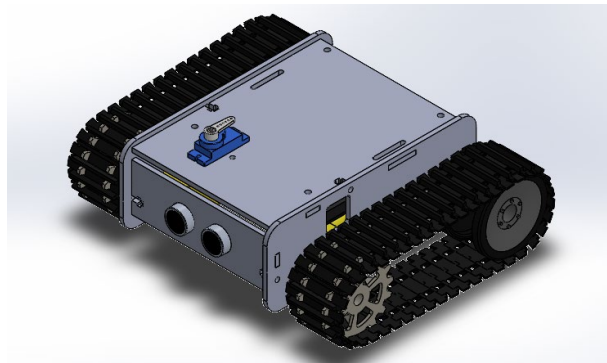


Ilustración 37: Vista isométrica de la parte móvil

8.3. Parte de brazo manipulador robótico

Para el diseño de esta estructura se parte de la que se ha preparado en el apartado de preparación previa 8.1.3. A partir de ahí, se desplaza la estructura de brazo en su conjunto hacia el suelo, se incluye una estructura de cabina y se diseña una estructura para el movimiento de la pala.

Toda la estructura en su conjunto debe ensamblar los tres servomotores que permiten el desplazamiento según sus 3 grados de libertad y el servomotor de la parte de la pala o cuchara.

Para nombrar todas las piezas que forman este apartado se ha tomado una nomenclatura numérica partiendo de la base, pasando por la cabina y llegando a la pala, de forma que, por ejemplo, la base de la parte móvil corresponde con la pieza nº1 del manipulador. En particular, esta pieza se modifica a lo largo de todo el diseño al ser la base de toda esta estructura. No se nombrarán durante el proceso de diseño aquellas piezas que no se han modificado o que han sufrido cambios insignificantes respecto a su diseño original en el robot meArm.

8.3.1. Desplazamiento de la estructura

El objetivo de estos cambios es acercar al suelo el eje de los dos servomotores de la parte media del robot meArm, es decir, los que permiten el posicionamiento del efector final, en este caso la pala. Para ello se parte de la estructura preparada en el apartado 8.1.3. de preparación previa.

El motivo de esta decisión es que la estructura en el nuevo robot está en una posición mucho más alta que en la versión original meArm, ya que la estructura móvil junto con las ruedas ha elevado el conjunto considerablemente. Para llevar esto a cabo, se mantienen todas las distancias importantes entre ejes con respecto al robot meArm, de manera que no afecten a los cálculos cinemáticos del efector final. El resultado tras el ensamblaje de esta parte es el de la ilustración 38.

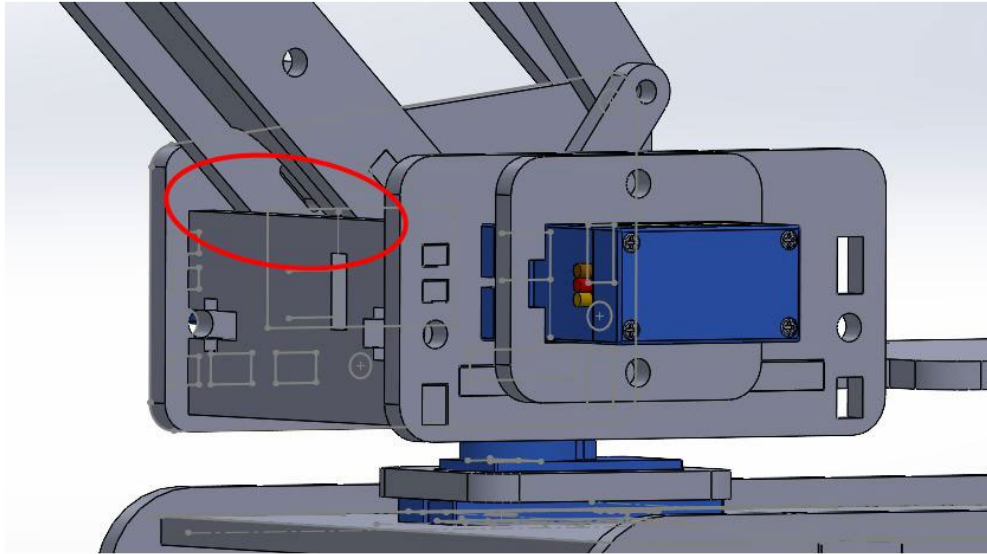


Ilustración 38: Resultado del desplazamiento de la estructura de la parte de manipulador robótico

Se observa que, con esta nueva disposición, el brazo al bajar colisiona con la estructura tal y como se señala en rojo en la ilustración 38. Por ello, se elimina esta pieza y se introduce una nueva barra, la pieza nº11 del manipulador, para cuyo ensamblaje se ajustan las piezas nº2 y nº3. Los cambios realizados respecto al modelo original se pueden observar en la comparativa entre las ilustraciones 39 y 40.

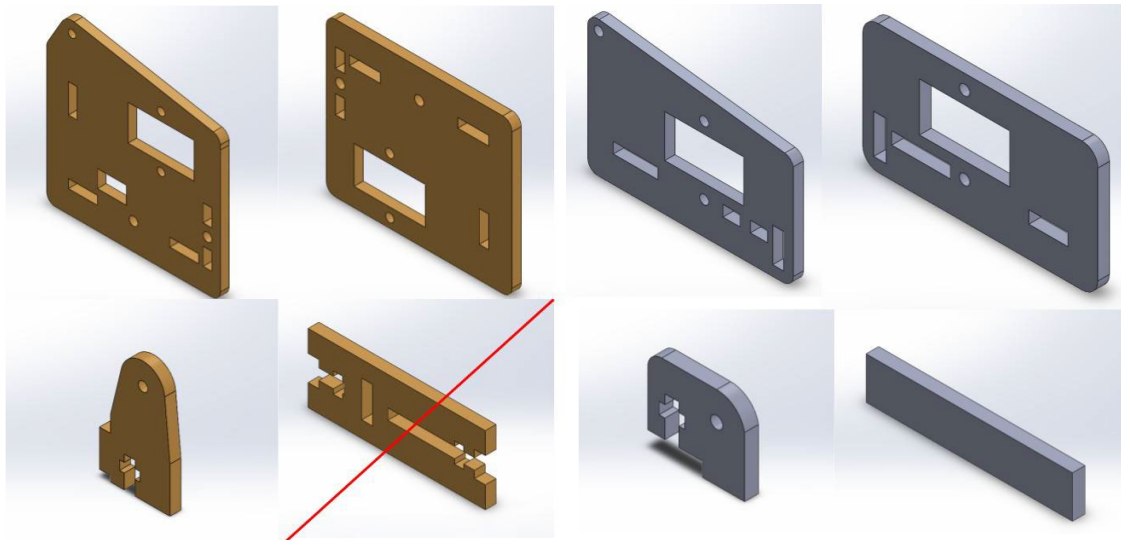


Ilustración 40: Vistas isométricas de las piezas originales nº2, nº3 y nº5 respectivamente, del robot meArm, afectadas en el desplazamiento de la estructura.

Ilustración 39: Vistas isométricas de las piezas nº2, nº3, nº5 y nº11 respectivamente, de la parte de robot manipulador, afectadas en el desplazamiento de la estructura.

Respecto a la ilustración 39, en el caso de las piezas nº2 y nº5, se trata de la versión definitiva. Sin embargo, para el caso de las piezas nº3 y nº11, estas se modifican de nuevo en el apartado 8.3.3. para dar lugar a la cabina.

8.3.2. Pala

Para el diseño de la estructura de la pala, se determina que tenga un servomotor el cual sea capaz de mover esta verticalmente, por lo que la estructura debe ensamblarlo.

En primer lugar, se analiza la configuración del robot meArm. Este, tal y como se detalló en el apartado 2.2., tiene configuración de robot paralelo para mantener el efector final en una misma orientación (horizontal en el caso del meArm), aunque se muevan los servomotores del brazo manipulador.

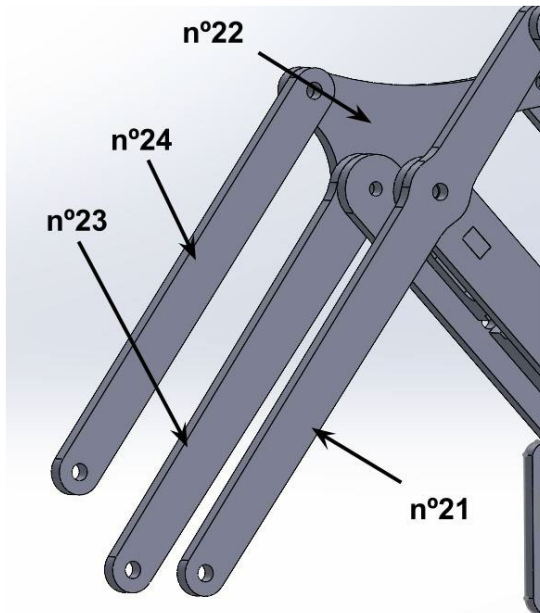


Ilustración 41: Piezas del extremo del manipulador robótico

estructura asociada a la pala se mueve con el cuerpo del servomotor, mientras que la parte que está fija es la manilla del servo. Esta se ha fijado a la pieza n°23. Para ello se ha modificado esta pieza, añadiéndole la ranura que se observa en la segunda imagen de la ilustración 42, donde va atornillada.

Para mantener la configuración de robot paralelo nombrada al principio del apartado, se ha creado la pieza n°25, la cual aprovecha el eje de la manilla del servomotor para apoyarse.

El movimiento del resto de la estructura se ha fijado al del cuerpo del servomotor y su función es estética. Para fijar el servomotor a la estructura, se ha utilizado una nueva pieza, la n°9. El diseño de esta pieza también ha sido influenciado por la cabina en el apartado 8.2.4.

Observando la ilustración 41, esto se consigue teniendo las circunferencias de los extremos vacíos de las piezas n°21 y n°23 concéntricas y manteniendo paralelas las piezas n°23 y n°24 mediante una pieza en sus respectivos extremos vacíos. Se quiere mantener esta configuración por lo que se parte de este punto.

Además, puesto que el movimiento de la pala se realiza verticalmente, el servomotor debe estar en posición horizontal al igual que los dos que se encargan de mover las piezas del brazo.

Con todas estas consideraciones, se realiza el diseño de la ilustración 42. Este diseño funciona de manera que la

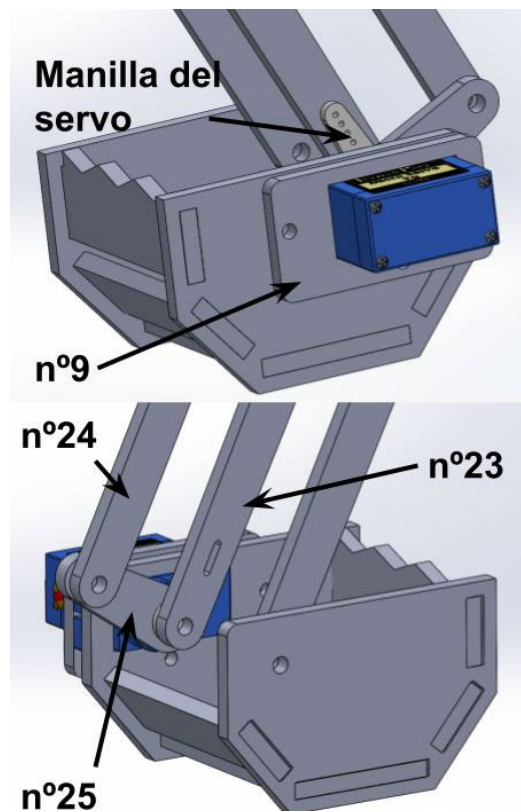


Ilustración 42: Diseño de pala desarrollado

Por último, la parte de atrás visible en la segunda imagen de la ilustración 42 se ha dejado abierta para facilitar el movimiento de la pala y que no colisione con la pieza nº24.

8.3.3. Cabina

El objetivo que tiene el diseño de esta estructura es ensamblar el módulo de cámara ESP32-CAM, además de cumplir con el requisito estético propuesto de representar la estructura de cabina de una máquina excavadora real. Por ello, se coloca al lado izquierdo del manipulador.

En primer lugar, se tienen en cuenta las dimensiones de dicho módulo. Estas son las de la ilustración 43 y son del vendedor "DFRobot", las cuales se ha comprobado su veracidad con el calibre.

La medida referente al espesor no viene proporcionada y varía según los elementos electrónicos de la placa. Esta se ha medido con el calibre y se ha determinado que es la forma más fácil para ensamblar el componente. La parte con siglas ANT de la ilustración 43 tiene un grosor de unos 2,5 mm y se encuentra en la parte inferior de la tarjeta, mientras que la cámara se encuentra en la parte superior (entre los pines) y tiene un grosor de 4 mm. De esta forma y comprobando que el saliente no colisiona con la parte de abajo del robot, se decide abrir una ranura en la pieza nº1 (base del manipulador) para dejar pasar únicamente la parte ANT y que de esta forma se mantenga vertical y fija la cámara.

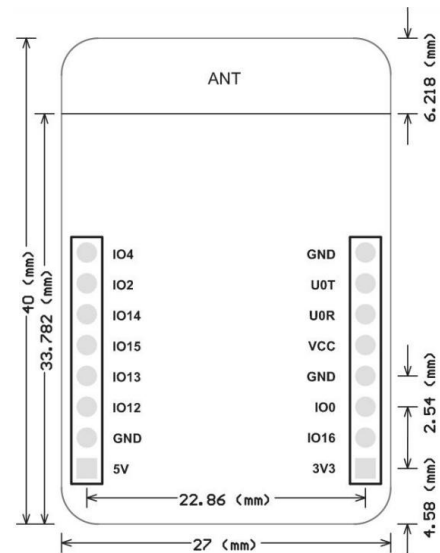


Ilustración 43: Dimensiones ESP32-CAM

Respecto a la estructura de la que se parte, dado que se tiene un servomotor en ese lado, cuya pieza de sujeción (nº14) colisionaría al alargar la pieza de la base de la parte del manipulador (nº1), se decide cambiarla por una pieza nº9 modificándola, de forma que un mismo diseño sea apto para esta estructura y la del apartado anterior de la pala. Dicha colisión se observa en la ilustración 44. Además, esto provoca una nueva edición de la pieza nº3.

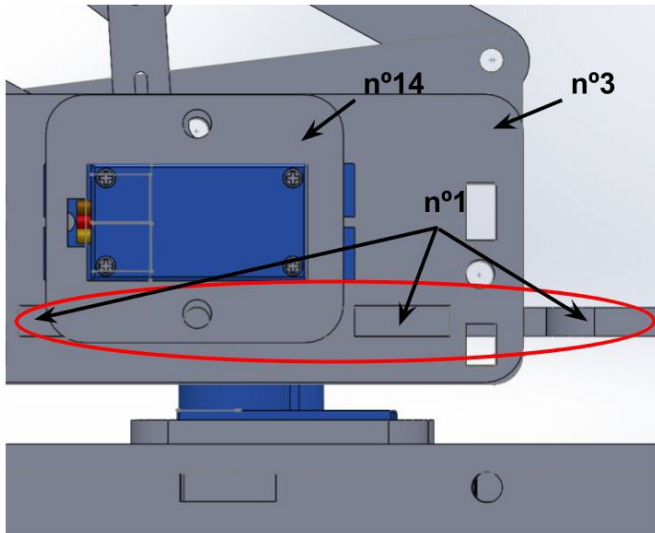


Ilustración 44: Proceso de diseño de la base del manipulador

Los agujeros del lateral derecho de la figura 3 que se han dejado y se observan en la ilustración 44, se aprovecharán para, a partir de estos, crear la estructura de la cabina.

Con todas estas consideraciones, se crea la estructura de cabina de la ilustración 45.

La pieza n°3 se ha modificado considerablemente teniendo en cuenta todo lo anterior, aumentando su altura y con una curva que asemeja la luna de la cabina de una máquina excavadora real. La pieza n°11

también se ha alargado para abarcar la nueva anchura y la pieza n°1 (de la base) se ha alargado también, dejando la ranura que se observa en la segunda imagen de la ilustración 45 para el módulo de la cámara.

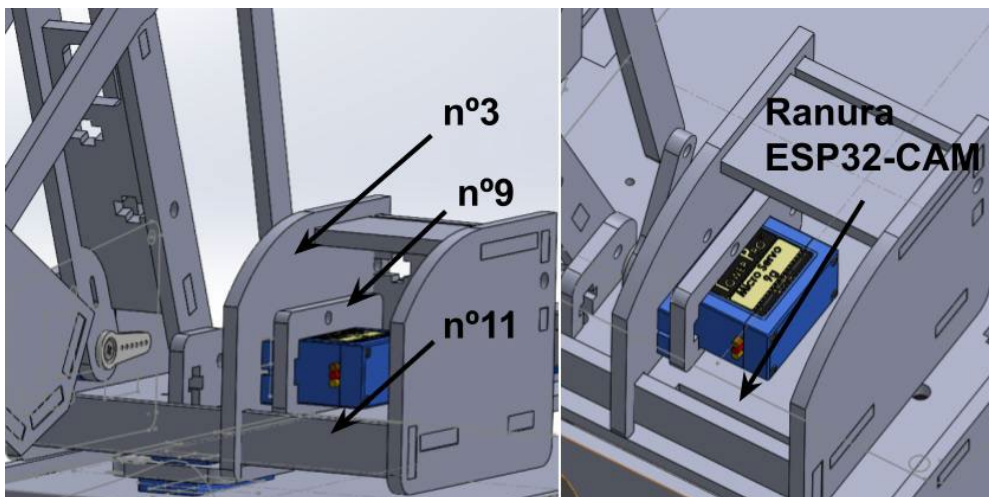


Ilustración 45: Estructura de cabina diseñada

8.4. Ensamblaje final

Para el ensamblaje final, puesto que se ha ido ensamblando durante el diseño, se añaden aquellos componentes que faltan como el módulo ESP32-CAM y se le da color al conjunto. Puesto que la estructura está preparada para su realización en corte láser, se le da apariencia de madera de arce pulido, opción dentro de las apariencias de SolidWorks. Esta es comúnmente utilizada en corte láser. El resultado final se representa en la ilustración 46.

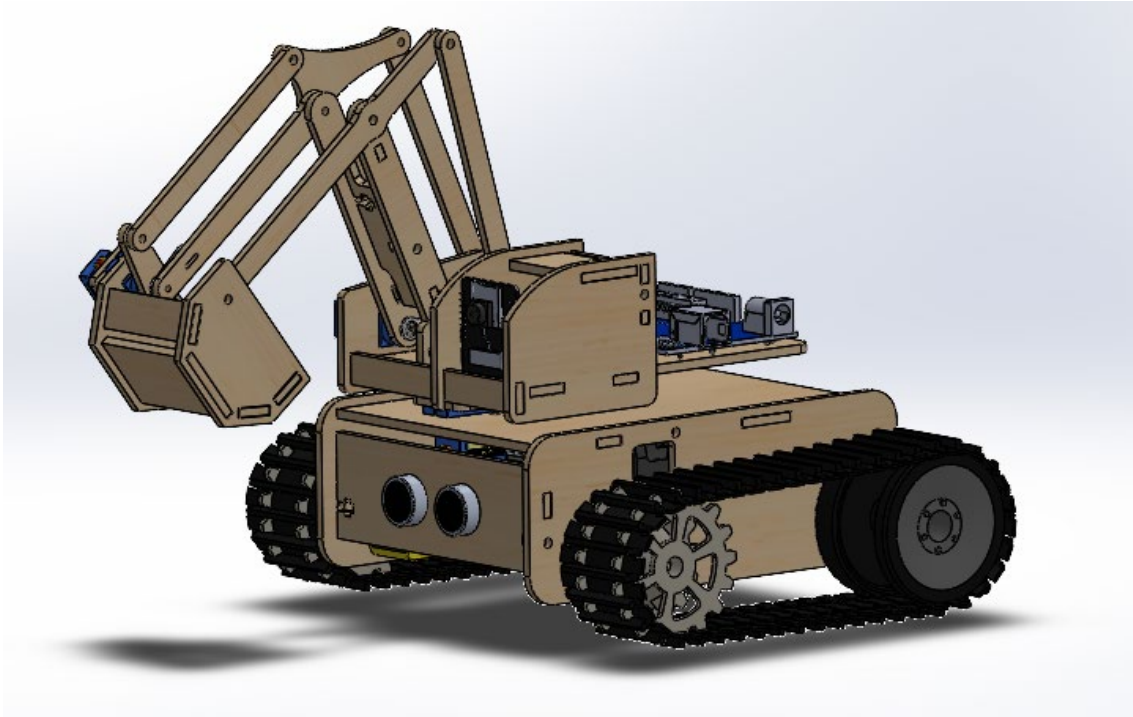


Ilustración 46: Resultado final del modelado de robot excavadora implementado

9. Exportación para la simulación en V-REP (CoppeliaSim)

El software V-REP CoppeliaSim permite la simulación de robots en 3D. Este es un software comúnmente utilizado tanto en principiantes como en profesionales y presenta varias ventajas frente a otros simuladores, entre las que se destaca sus varios métodos de programación (API, Visual Studio...), soporte de varios lenguajes (Lua, C++, Matlab ...) y gratuidad.

Sus métodos de programación se basan en las relaciones entre eslabones y articulaciones. En el caso del modelo desarrollado, será necesario establecer estas relaciones en el programa además de ser importado.

Desde SolidWorks, para la exportación del modelo al software de simulación V-REP CoppeliaSim, se utiliza una herramienta instalable en este. Se trata de la herramienta "SolidWorks to URDF Exporter", abreviada como "SW2URDF". Esta es proporcionada por un usuario de GitHub a través del enlace https://github.com/ros/solidworks_urdf_exporter.

GitHub se trata de una plataforma abierta a todos los usuarios, para creación de proyectos y aplicaciones. Se trata de un repositorio creado para alojar código abierto de aplicaciones y herramientas, de forma que otros usuarios puedan acceder a él, utilizarlo y crear nuevas ramificaciones y proyectos.



Ilustración 47: Logotipo de Github

Una vez descargada esta herramienta y antes de comenzar, se debe poner el robot en una posición sencilla para su programación. Es por ello por lo que se alinean las estructuras de los brazos de igual forma que en el robot meArm, es decir, en posición de forma que los servomotores formen 90 grados, a excepción del servomotor de la pala. De esta es más sencillo, se conocen los límites de las articulaciones y, en consecuencia, el área de trabajo del manipulador ya que es igual que en el robot meArm. El resultado es el de la ilustración 49.

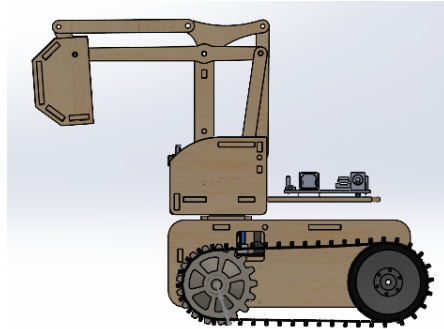


Ilustración 49: Posición del robot para su programación

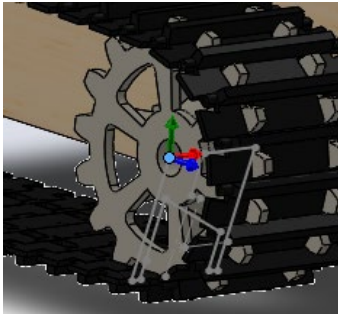


Ilustración 48: Ejemplo de Sistemas de Referencia definidos

A continuación, se definen los sistemas de referencia a partir de los cuales se ejecutará esta herramienta (ilustración 48). Estos se definen en el caso de las ruedas en el eje de giro y en las articulaciones, justo en la articulación.

Posteriormente, en el buscador de SolidWorks se busca “urdf” para localizar la herramienta “Exportación a URDF” instalada y ejecutarla. Esta herramienta funciona de forma que se le debe especificar la jerarquía de los eslabones dentro de SolidWorks para que, de forma automática o semiautomática, genere las relaciones entre eslabones y articulaciones en formato URDF para poder abrirse en CoppeliaSim. Cabe destacar que esta herramienta presenta limitaciones en la representación de robots como el de nuestro caso. Esta es que solo permite definir una relación de jerarquía escalonada, no siendo así la de nuestro manipulador robótico. Es decir, en nuestro caso, con esta herramienta no se podrá representar la estructura de robot paralelo del manipulador y se tendrá que terminar de definir en el propio CoppeliaSim.

La interfaz de la herramienta con un ejemplo de uso, las relaciones establecidas mediante la herramienta SW2URDF en nuestro caso y los sistemas de coordenadas utilizados, son los siguientes:

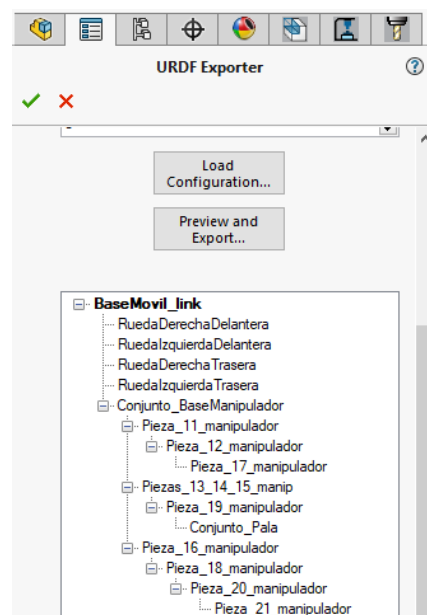


Ilustración 50: Interfaz y jerarquía establecida en SW2URDF

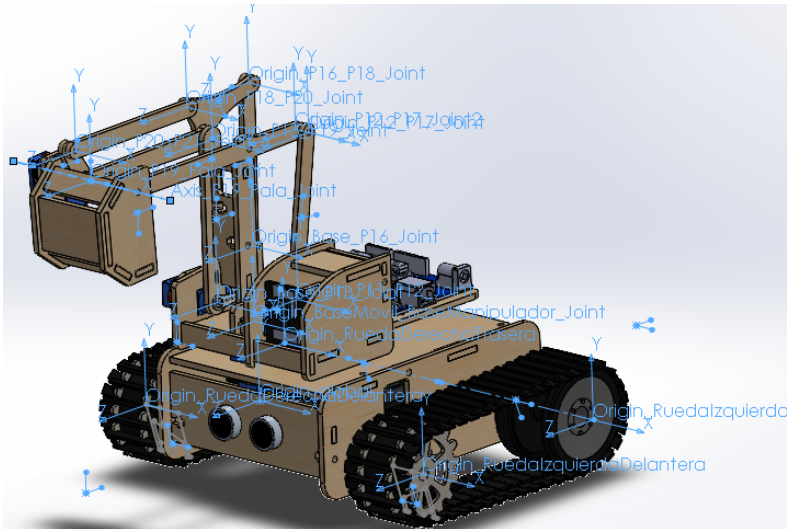


Ilustración 51: Sistemas de coordenadas utilizados

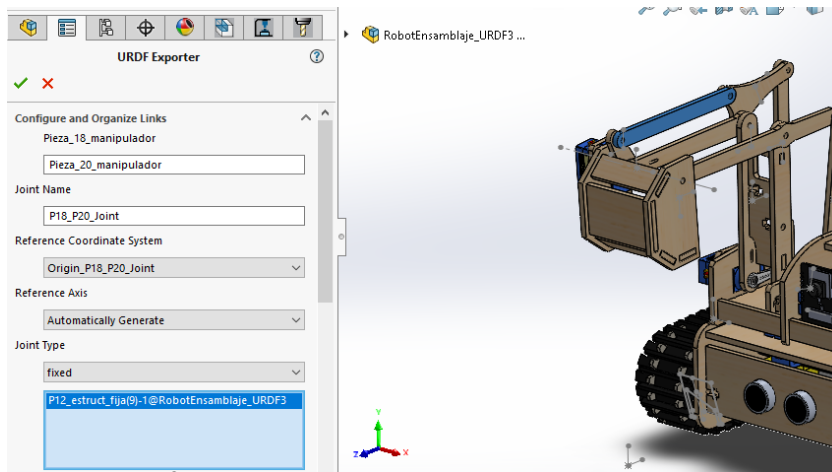


Ilustración 52: Ejemplo de uso de SW2URDF

De los elementos de la parte móvil, se toman como base (BaseMovil_link en la ilustración 50) sus elementos electrónicos (motores DC, ...) y todas sus estructuras (base, laterales y frontal) incluidas las orugas, excluyendo las 4 ruedas. El objetivo de esta decisión es simplificar el sistema de movimiento, ya que la animación de las orugas únicamente tendría fines estéticos. El movimiento únicamente se atribuye a las ruedas motrices. Las ruedas conducidas son imitadoras del movimiento de las anteriores.

La relación de jerarquía del manipulador de nuestro robot es la misma que la del robot meArm hasta el comienzo del efector final. En el caso de la pala es más sencillo y se ha podido realizar con la herramienta SW2URDF anteriormente citada: se define toda la estructura como un solo eslabón (piezas nº22 a la nº27 del manipulador) y se define una sola articulación a partir del sistema de coordenadas ubicado en la intersección de la pieza del manipulador nº19 con el eje del servomotor de la pala (ilustración 53).

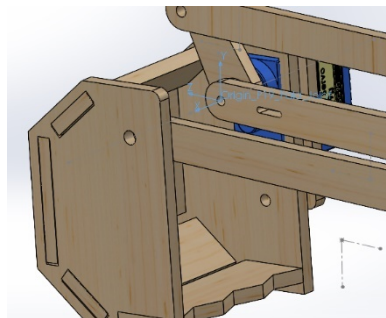


Ilustración 53: Sistema de coordenadas de la pala

Con todo esto en cuenta, se ejecuta la herramienta. Al acabar el proceso se abre una ventana con opciones de definición de límites (velocidad, ángulos, etc.), sin embargo, esto ya puede hacerse en CoppeliaSim por lo que se omite.

Dentro de CoppeliaSim, se abre el archivo en: Plugins – URDF Import. El resultado es el de la ilustración 54. La mayoría de los colores se han perdido en la importación y al haber definido conjunto de piezas como eslabones, no es posible aplicarle a cada elemento su color por lo que se deja como está y se prescinde de esta estética.

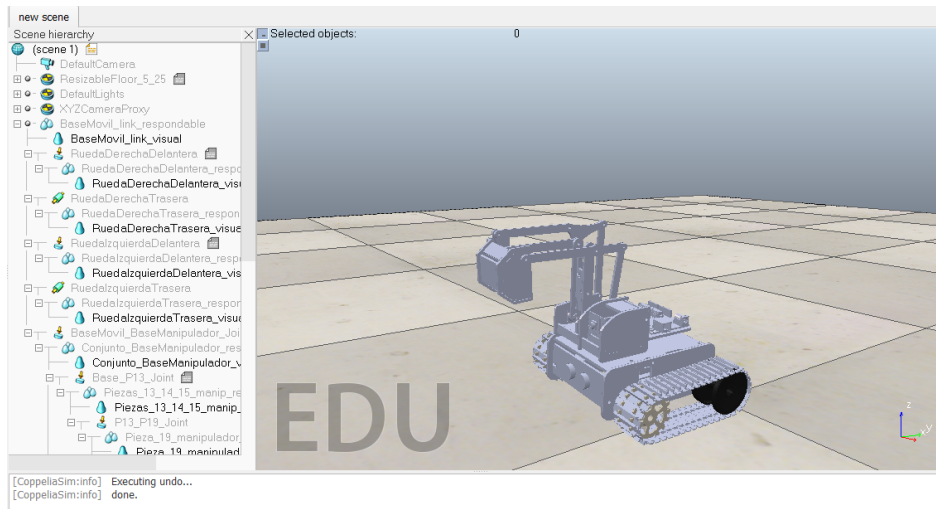


Ilustración 54: Resultado de exportación en CoppeliaSim

Por último, teniendo en cuenta las relaciones de jerarquía del robot meArm hasta el efector final de la ilustración 55, se puede terminar de definir las articulaciones que faltan entre los eslabones.

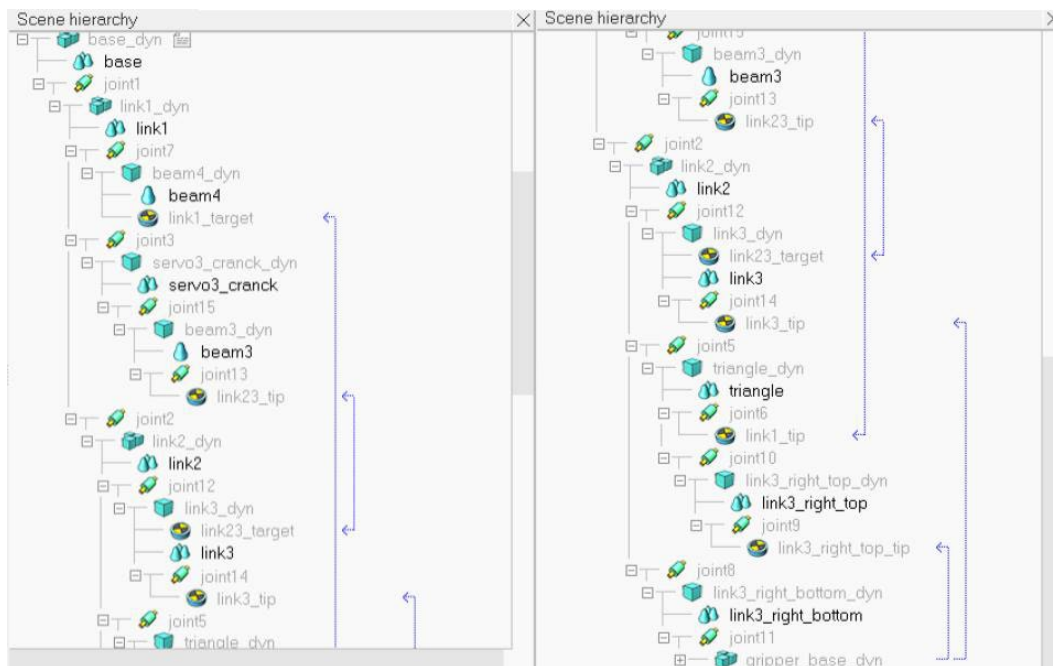


Ilustración 55: Jerarquía del robot meArm hasta el eje del efector final (pala en este caso)

Llegados a este punto, desde la ventana *Scene Object Properties – Show Dynamic Properties Dialog* se le pueden aplicar propiedades dinámicas al robot para una simulación más realista.

La creación de una escena se omite puesto que ya se realiza un montaje del robot y una simulación real.

10. Programación

La programación del robot excavadora puede dividirse en 2 partes independientes. Por un lado, se programa el módulo ESP32-CAM con fines de control y por otro el resto de los componentes que controla la placa Arduino UNO.

10.1. ESP32-CAM

Para programar el módulo de la cámara, se necesita acceder a este a través de conexión TTL. Esto es *transistor to transistor logic*, lo cual es una tecnología de construcción de circuitos integrados digitales, basada en el uso de transistores bipolares. Es sucesora de otras tecnologías como la RTL (*resistor transistor logic*) y DTL (*diode transistor logic*), ofreciendo mayor velocidad y mejor relación señal-ruido respecto a estas.

Para ello, en el mercado existen módulos de conexión USB a TTL a bajo precio como el de la ilustración 56. Sin embargo, si se quiere prescindir de este componente, se puede utilizar también un Arduino NANO o Arduino UNO ya que las placas Arduino también tienen puertos TTL.

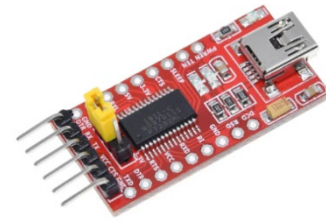


Ilustración 56: Módulo USB a TTL

En nuestro caso se dispone de un Arduino UNO, por lo que se realiza con este. Para ello, se deben conectar estos puertos TTL de Arduino, en concreto RX (recepción) y TX (transmisión) a los puertos U0R (recepción) y U0T (transmisión) de la ESP32-CAM. Para trabajar con Arduino en este modo, también se debe conectar el pin RESET a masa (GND) y en la ESP32-CAM el primer pin digital GPIO 0 a masa (GND). Por último, se conecta la alimentación de 5V entre ambas placas. El diagrama de conexiones corresponde al de la ilustración 57.

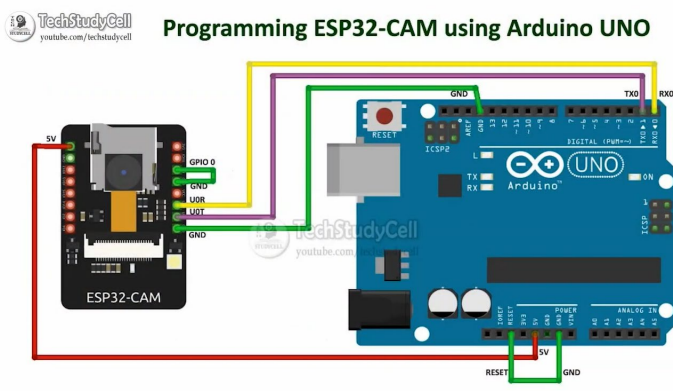


Ilustración 57: Diagrama de conexiones Arduino UNO y ESP32-CAM

Una vez conectado de esta forma, se procede a descargar el controlador de tarjetas ESP32 desde la interfaz de Arduino IDE. Para ello, en primer lugar, se añade la dirección de descarga en: Preferencias – Gestor de URLs adicionales de tarjetas - https://dl.espressif.com/dl/package_esp32_index.json .

Tras ello, se descarga la librería ESP32 desde el buscador en: Herramientas – Administrar bibliotecas.

Con este módulo de cámara se pueden realizar múltiples aplicaciones, por ejemplo, de visión artificial mediante programación con Python. Sin embargo, para este proyecto su función únicamente será de supervisión para un mejor control, por lo que se utiliza desde Arduino IDE el script que trae esta librería por defecto.

Para ello, se accede a este desde el menú: Archivo – Ejemplos – ESP32 – CameraWebServer. Una vez abierto, se selecciona del código comentado el modelo adquirido de ESP32-CAM, en nuestro caso el modelo de la línea 17 en la ilustración 58, y en las líneas 22 y 23 se introduce el ID y contraseña Wifi. Por último, en herramientas, se seleccionan las opciones de la ilustración 59.

```

10 // Select camera model
11 //#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
12 //#define CAMERA_MODEL_ESP_EYE // Has PSRAM
13 //#define CAMERA_MODEL_MSSTACK_PSRAM // Has PSRAM
14 //#define CAMERA_MODEL_MSSTACK_V2_PSRAM // M5Camera version B Has PSRAM
15 //#define CAMERA_MODEL_MSSTACK_WIDE // Has PSRAM
16 //#define CAMERA_MODEL_MSSTACK_ESP32CAM // No PSRAM
17 #define CAMERA_MODEL_AI_THINKER // Has PSRAM
18 //#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
19
20 #include "camera_pins.h"
21
22 const char* ssid = "XXXXXXXXXX";
23 const char* password = "XXXXXXXXXXXX";
24

```

Ilustración 58: Código ESP32-CAM CameraWebServer

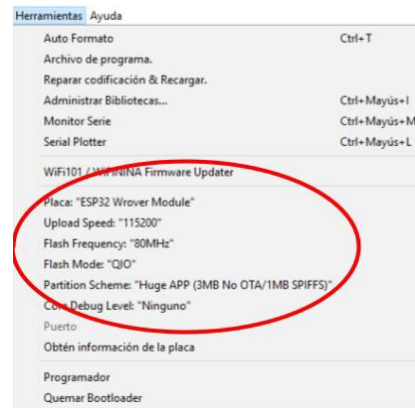


Ilustración 59 Ventana de herramientas para programa ESP32-CAM

Se sube el código y se pulsa, mientras se compila, el botón RESET de la placa ESP32-CAM. Tras 1 minuto que tarda en subirse, se abre el monitor serie y se pulsa el botón RESET de la eSP32-CAM de nuevo. Aparece en ese momento una dirección IP, la cual al abrirla se muestra la retransmisión de la cámara junto con la interfaz de la ilustración 60.

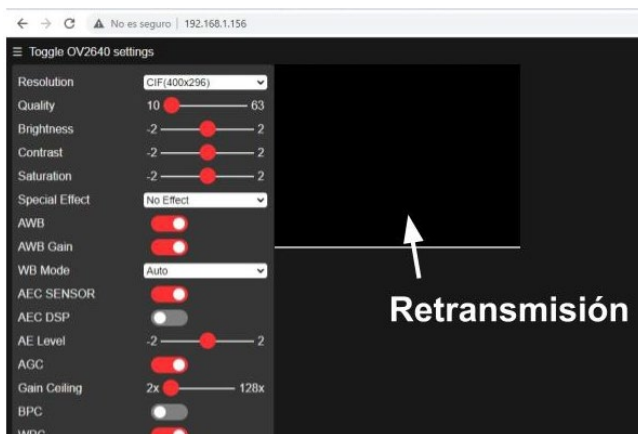


Ilustración 60: Interfaz de aplicación ESP32-CAM

De las opciones disponibles en la interfaz, únicamente se pulsa el botón *Start Stream*, ubicado al final. En ese momento, ya se pueden desconectar los cables y conectar la cámara mediante alimentación al robot para realizar un seguimiento de este en tiempo real.

10.2. Arduino IDE

Para realizar la programación del robot excavadora desarrollado, previamente se deben tener en cuenta los límites de sus articulaciones y sus dimensiones, así como su cinemática inversa y directa.

Para las articulaciones, se tiene en cuenta la posición de cada servomotor y su eje de giro. Las articulaciones se representan en la ilustración 61. En ella, los servomotores de los ejes de las articulaciones 1, 2 y 3 se encuentran a 90° y el del eje 4 de la pala a 85°.

Teniendo en cuenta el modelo diseñado, los componentes electrónicos y la simulación que se realiza a nivel del suelo, se define el **espacio de trabajo** del robot excavadora gracias al ensamblaje realizado en SolidWorks y experimentando con las relaciones de posición de ángulos entre las distintas piezas. El resultado es el siguiente:

La articulación 1 gira entre $q_1 = 65^\circ$ y $q_1 = 115^\circ$.

La articulación 2 gira entre $q_2 = 90^\circ$ y $q_2 = 175^\circ$.

La articulación 3 gira entre $q_3 = 0^\circ$ y $q_3 = 95^\circ$.

El ángulo de la articulación 4 está influenciado por la combinación de los otros 2 ángulos, ya que la manilla del servomotor se sujeta en la estructura de paralelogramo del brazo. Esto trae como consecuencia que la definición de su espacio de trabajo resulta compleja para que no colisione con la estructura. Por ello, en la aplicación que se desarrolla más adelante, simplemente se comprueba previamente que no colisione.

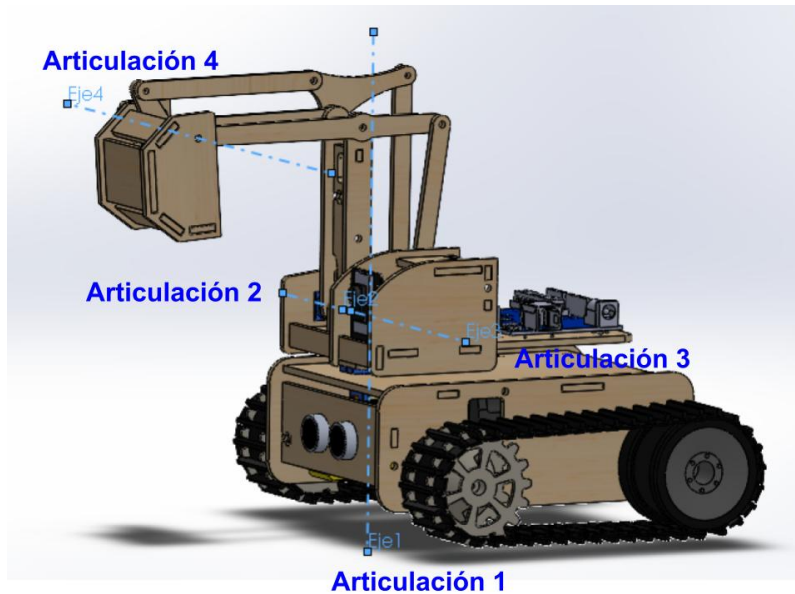


Ilustración 61: Articulaciones del robot excavadora

Las dimensiones del manipulador del robot son las de las ilustraciones 62 y 63 junto con la tabla de valores de la ilustración 64.

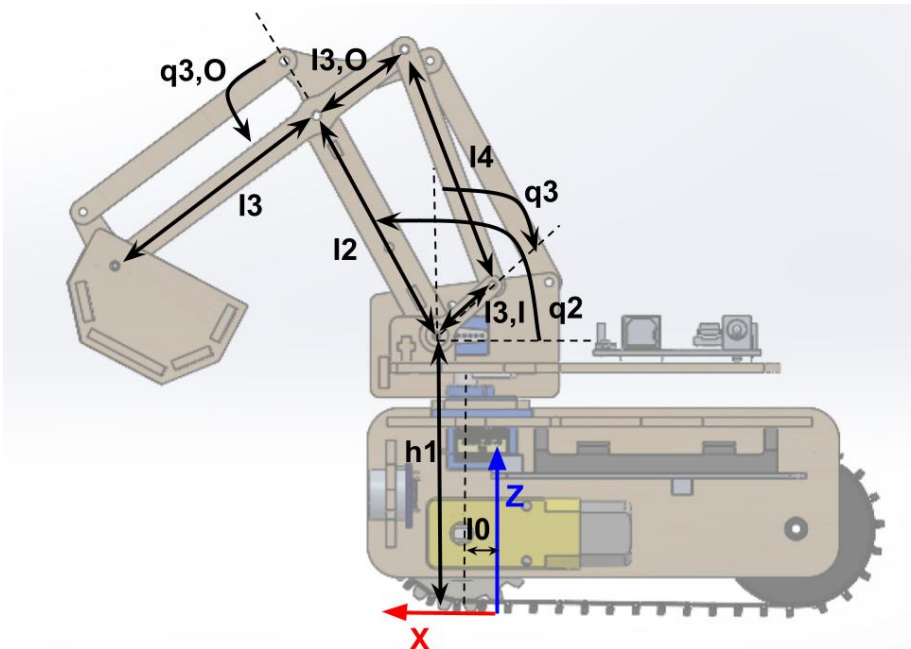


Ilustración 62: Dimensiones del manipulador del robot excavadora 1 ($q_1=90^\circ$, $q_2 > 90^\circ$ y $q_3 < 90^\circ$).

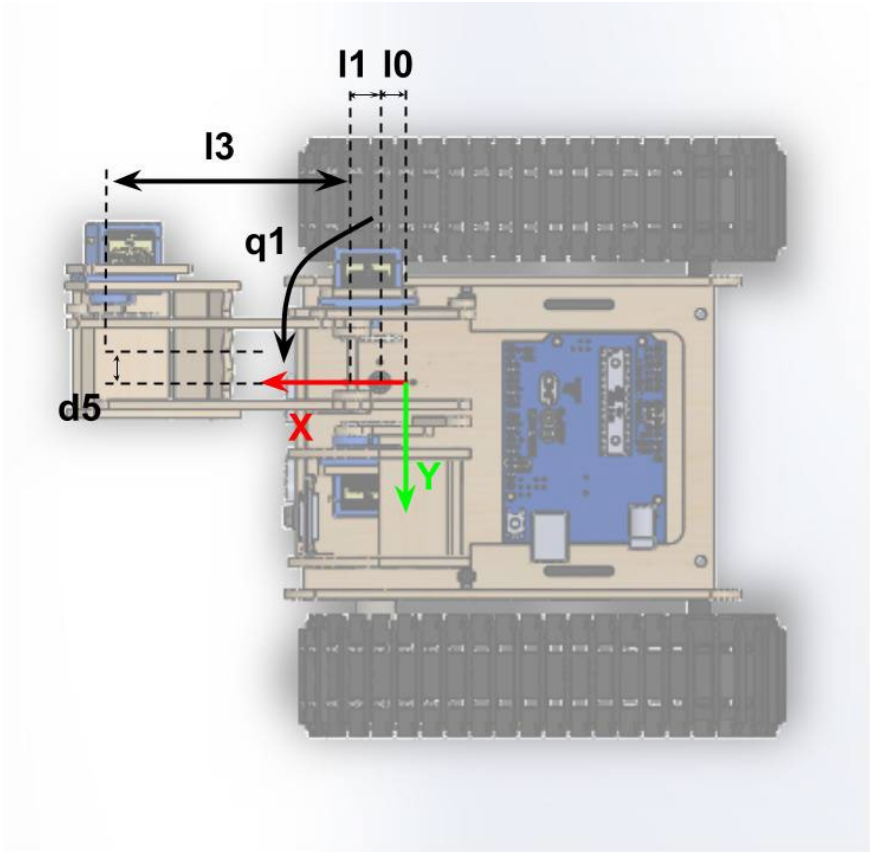


Ilustración 63: Dimensiones del manipulador del robot excavadora 2 ($q_1=q_2=q_3=90^\circ$)

Dimensiones	mm
l_0	5
h_1	87.5
l_1	15.1

l_2	80
l_3	80
$l_{3,I}$	23.9
$l_{3,O}$	35
l_4	80
d_5	10.7
l_5	0

Ilustración 64: Tabla de valores de las dimensiones

Se observa que respecto a las **dimensiones** del modelo del robot meArm (anexo 1), se ha anulado la medida l_5 puesto que esta era consecuencia de la pinza y del punto objetivo (p_T) de esta. También se han modificado h_1 , ya que el suelo donde se sitúa el sistema de coordenadas está a distinta altura, y d_5 puesto que se ha decidido que llegue a una posición más central en la pala.

Para la **cinemática inversa**, en comparación al robot meArm (anexo 1), el punto objetivo (p_T) ahora corresponde al punto al que se llega desde l_3 y d_5 en la ilustración 65. Teniendo en cuenta además el sistema de coordenadas de la ilustración 63, el cálculo del punto de muñeca (p_w^0) desde el punto objetivo (p_T) es lo único que cambia en estos cálculos. Este queda de la siguiente forma expresado en coordenadas cartesianas XYZ:

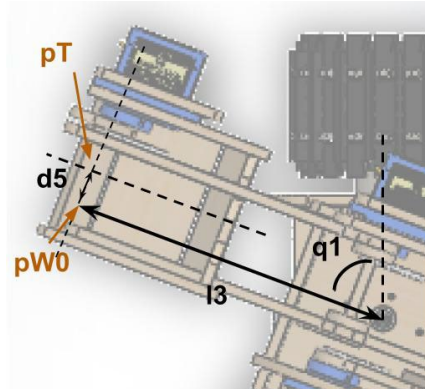


Ilustración 65: Ubicación p_w^0 y p_T

$$p_w^0 = p_T + \begin{bmatrix} d_5 \cos(q_1) \\ -d_5 \sin(q_1) \\ 0 \end{bmatrix}$$

En el resto de los cálculos, puesto que $l_5=0$, se anula en las ecuaciones en las que aparece y no es necesario modificarlas. Lo anterior es aplicable de igual forma a la **cinemática directa**.

Se ha decidido que de entre las distintas posibilidades de movimientos estudiados en el robot meArm (anexo 1), entre los que se encuentran movimiento de ejes (moveAbsJ), movimiento punto a punto (moveJ) y movimiento lineal (moveL), se utiliza el movimiento punto a punto. Este usa además la función de movimiento de ejes. La razón de esta decisión es que en este robot excavadora no se considera necesario usar varios tipos de movimiento ni gran exactitud en la trayectoria de estos como para implementar un movimiento lineal.

Con todas estas consideraciones, la programación que se lleva a cabo se basa en el esquema de funcionamiento de la ilustración 66.

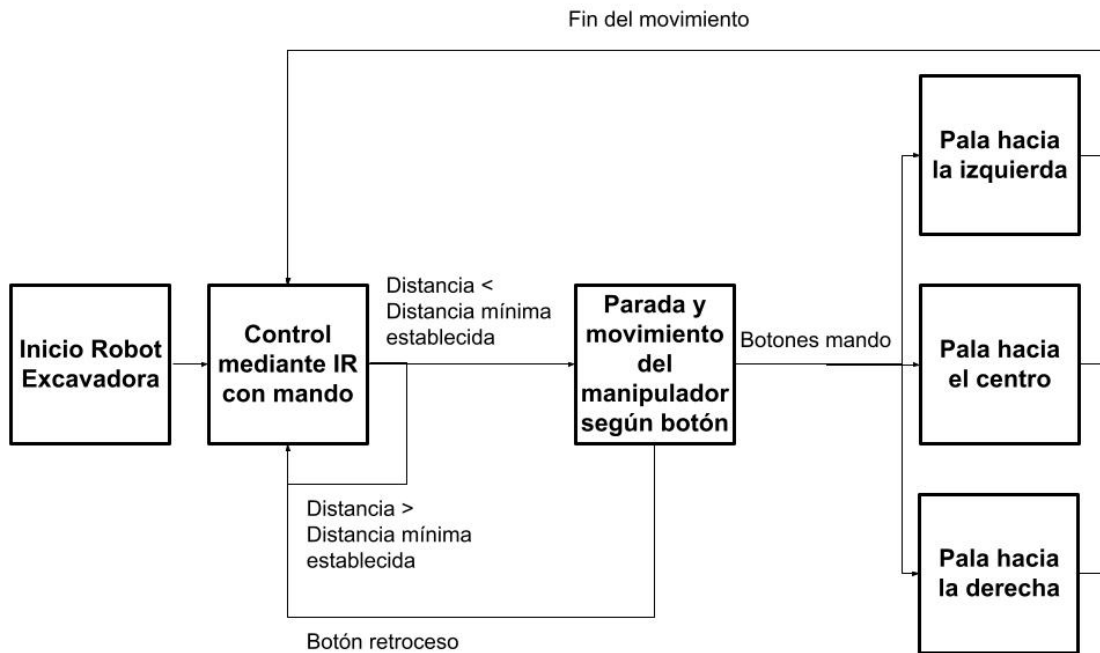


Ilustración 66: Esquema de funcionamiento establecido

En esta aplicación desarrollada, el robot funciona de manera que se tiene un control manual de este en todo momento. Sin embargo, cuando el sensor de ultrasonidos detecta una distancia por debajo de un mínimo (establecido en 5 cm), se detiene dando opción a que el manipulador haga un movimiento a uno de los 3 sitios especificados. En cada sitio y según el botón pulsado, se tiene la opción de carga o de descarga. El usuario debe elegir en cada ocasión qué tipo de movimiento realizar y a qué sitio según sea lo más conveniente en esa ocasión. En caso de no estar seguro del movimiento, mediante el botón de retroceso configurado del mando, se podrá retomar el control del robot para un mejor posicionamiento.

Las medidas de distancia se obtienen del sensor HC-SR04, el movimiento de la parte móvil mediante los 2 motores DC y el de la parte del manipulador con los 4 servomotores.

Para programar las distintas opciones de movimiento del manipulador, es necesario saber qué ángulo de los servomotores se tiene para unas coordenadas cartesianas XYZ, y si estos entran dentro del área de trabajo definida anteriormente. Para ello, se dispone de la cinemática inversa del robot.

De igual forma, para fijar la posición inicial del manipulador y que esta se asemeje lo máximo posible a una excavadora, es necesario el caso contrario, es decir, dados unos ángulos, conocer las coordenadas XYZ. Para ello se dispone de los cálculos de cinemática directa del robot meARm (anexo 1). Tal y como se ha descrito con

anterioridad al principio de este apartado, estos cálculos sirven también para este robot cambiando los parámetros nombrados.

Ambos métodos (cinemática inversa y directa), se han implementado en 2 scripts de MATLAB (anexo 3). Funcionan de forma que se introducen las coordenadas de un tipo, angulares o cartesianas, y devuelve el contrario.

Respecto a las coordenadas de la **componente del eje X**, se tiene en cuenta la distancia entre el sistema de coordenadas y la lente del sensor HC-SR04 (ilustración 67) y que la distancia mínima establecida en la que el robot frena cuando detecta un objeto es de 50 mm. Por ello, cuando se llegue a la situación de frenado automático, el objeto se encontrará a $X=40.3$ mm del origen del sistema de coordenadas. Si se tiene en cuenta la medida de la ilustración 68 puesto que p_T se encuentra en el eje del servomotor de la pala, la distancia mínima en X a la que se debe enviar el manipulador es $X_{mín} = 5(10) + 30.3 + 43.35 + 50 = 128.65$ mm.

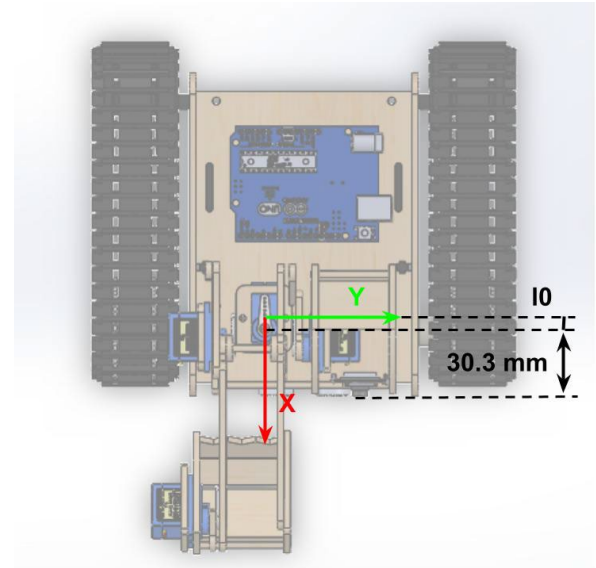


Ilustración 67: Distancia entre el sistema de coordenadas y la lente del sensor HC-SR04

Respecto a las coordenadas de la **componente del eje Z**, se tiene en cuenta que desde el eje 4 de la pala, donde se encuentra p_T , al extremo de la pieza n°22 del manipulador, hay una distancia de $Z=43.35$ mm (ilustración 68). Esta es la altura mínima que se debe poner en la coordenada cartesiana de la componente Z respecto al sistema de coordenadas, para permitir el giro de la pala sin colisionar con el suelo. Por ello $Z_{mín} = 43.35$ mm.

La **componente del eje Y** no tendrá mayor limitación que la ya establecida mediante los ángulos de los servomotores en el espacio de trabajo.

Por último, contando con el ensamblaje de SolidWorks para recrear las posiciones exactas mediante relaciones de posición de ángulos, se calculan las posiciones de los 3 movimientos (izquierda, derecha y centro).

Tal y como se describió al principio de este apartado, el espacio de trabajo del servomotor de la pala no es fácilmente definible por lo que en cada movimiento se ha comprobado que la estructura de la pala no colisione con el resto de las barras mediante la representación de todas las posiciones de los movimientos en

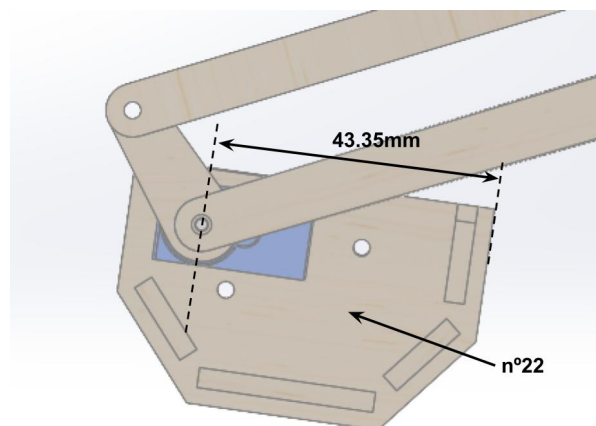


Ilustración 68: Distancia entre el eje del servomotor de la pala y extremo de la pieza del manipulador n°22.

SolidWorks. Por ello, se han definido estos valores para dicho servomotor:

Pala cerrada: $q_4=50^\circ$.

Pala abierta: $q_4=125^\circ$.

Cada movimiento de carga de pala se ha determinado que tenga 2 posiciones en 2 puntos distintos en cada sitio, de manera que desde la posición inicial se va hacia la primera en la que se abre la pala, y desde esta, a la segunda donde se cierra la pala. Tras esto, se vuelve a la posición inicial.

Posición inicial del robot: Coordenadas cartesianas: $Pt_0 = \{130.24, -10.7, 118.32\}$.
Ángulos en Pt_0 : $q_1 = 90^\circ$, $q_2 = 120^\circ$, $q_3 = 60^\circ$, con pala cerrada ($q_4 = 50^\circ$).

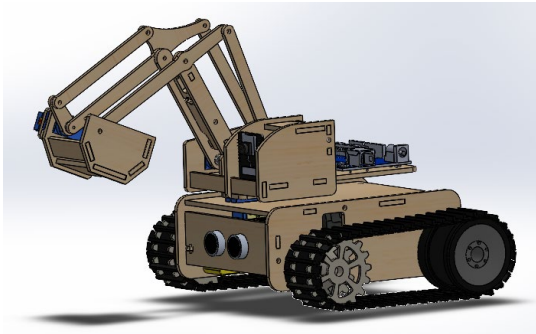


Ilustración 69: Posición inicial del robot Pt_0 con pala cerrada

Dichos puntos en cada sitio son los siguientes:

Movimiento de carga de pala hacia el centro:

Coordenadas cartesianas: $Pt_1 = \{160, -10.7, 53\}$. Ángulos en Pt_1 : $q_1 = 90^\circ$, $q_2 = 173.25^\circ$, $q_3 = 94.16^\circ$.

Coordenadas cartesianas: $Pt_2 = \{135, -10.7, 53\}$. Ángulos en Pt_2 : $q_1 = 90^\circ$, $q_2 = 158.02^\circ$, $q_3 = 41.21^\circ$.

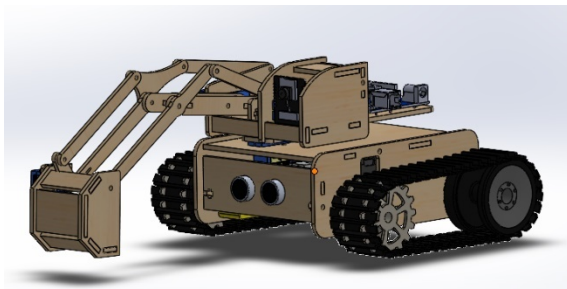


Ilustración 70: Posición Pt_1 con pala abierta

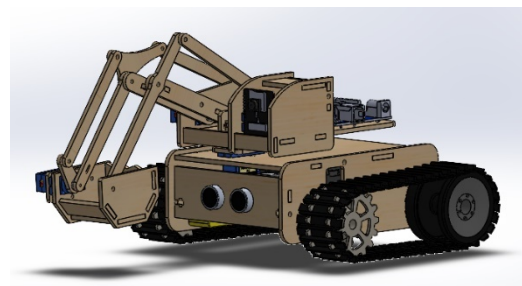


Ilustración 71: Posición Pt_2 del robot con pala cerrada

Movimiento de carga de pala hacia la izquierda:

Coordenadas cartesianas: $Pt_3 = \{143.4, 83.82, 53\}$. Ángulos en Pt_1 : $q_1 = 125^\circ$, $q_2 = 173.25^\circ$, $q_3 = 94.16^\circ$.

Coordenadas cartesianas: $Pt_4 = \{123.1, 69.64, 53\}$. Ángulos en Pt_2 : $q_1 = 125^\circ$, $q_2 = 158.02^\circ$, $q_3 = 41.21^\circ$.

Movimiento de carga de pala hacia la derecha:

Coordenadas cartesianas: Pt5 = {146.78, -77.92, 53}. Ángulos en Pt1: q1 = 65°, q2 = 173.25°, q3 = 94.16°.

Coordenadas cartesianas: Pt6 = {124.36, -67.47, 53}. Ángulos en Pt2: q1 = 65°, q2 = 158.02°, q3 = 41.21°.

Para los movimientos de descarga de pala, se utiliza únicamente un punto de los ya definidos para cada sitio. El efector final irá desde la posición inicial en Pt0 a uno de estos puntos según el sitio seleccionado, abrirá la pala, la cerrará y terminará volviendo a la posición inicial en Pt0. Dichos puntos son los siguientes:

Movimiento de descarga de pala hacia el centro: Pt1.

Movimiento de descarga de pala hacia la izquierda: Pt3.

Movimiento de descarga de pala hacia la derecha: Pt5.

Por último, se asignan los botones del mano a los movimientos. Para el control manual (distancia mayor que la mínima establecida), se utilizan 5 botones del mando para los movimientos básicos de avance, retroceso, giro a la izquierda, giro a la derecha y parada. Para el control del manipulador una vez (distancia menor a la mínima establecida), puesto que se tienen 3 sitios y 2 movimientos en cada sitio, es necesario utilizar 6 botones del mando. El control queda de la siguiente manera:



Ilustración 72: Control establecido con mando de IR

11. Montaje y recreación de la escena programada

En primer lugar, para la **fabricación** de las piezas, se ha tomado como material de fabricación DM (material de fibras de densidad media). No todo el robot en su conjunto se ha fabricado de este material puesto que se comenzó a fabricar en la universidad con

los recursos disponibles y se terminó fabricando de este material en una empresa a parte.

Los **tornillos** utilizados son de métricas M2x6 (rosca), M3x6, M3x12, M3x25 y M3x30 (pasantes). Estos se utilizan en los siguientes casos de forma general: los tornillos M2x6 se utilizan para fijar las manillas de los servomotores a las piezas de la estructura. Los tornillos M3x6 se utilizan para fijar el Arduino, las piezas de la estructura entre sí y para las articulaciones de la parte del manipulador. Los tornillos M3x12 se utilizan en articulaciones que encajen 3 eslabones y en las de los servomotores. Los tornillos M3x25 se utilizan en la fijación de la batería (el soporte con las pilas) y los tornillos M3x30 en la fijación de los motores DC a las estructuras laterales.

Para los **cables**, de forma general para las conexiones se utilizan cables macho-macho. También son necesarios al menos 9 cables macho-hembra (para alargar conexiones). Además, se recomienda utilizar de todas las longitudes, tanto cortos como de media longitud y largos.

El **montaje de cada componente y pieza** se realiza siguiendo los pasos definidos en el apartado 8 de Diseño y ensamblaje y teniendo conocimientos previos de la estructura del robot meArm, disponible de forma gratuita en varios repositorios online.

El resultado final del montaje es el de la ilustración 73.

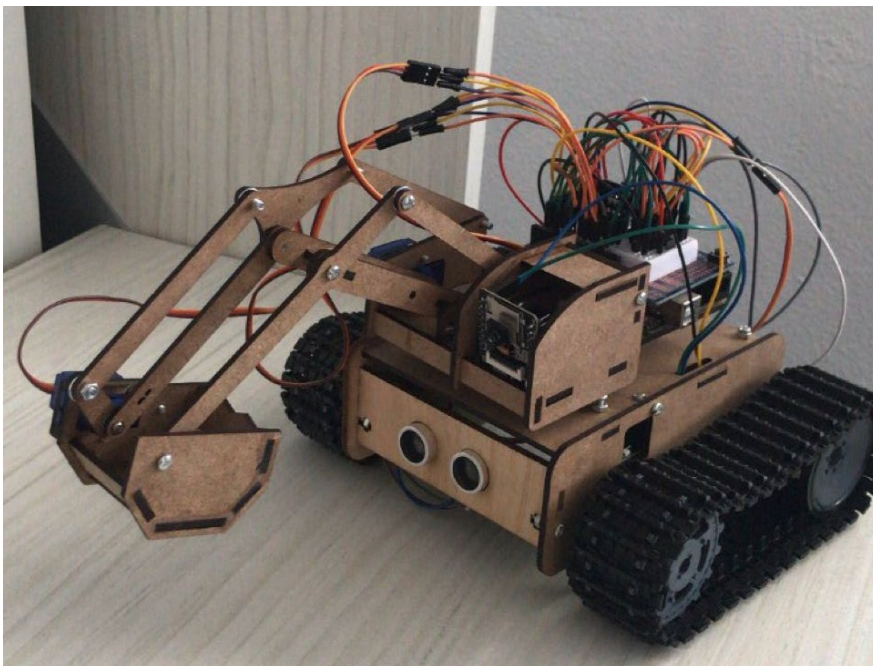


Ilustración 73: Resultado final del montaje

Un **ejemplo de escena y de uso de la programación** llevada a cabo en el apartado 10 de programación está disponible en el siguiente enlace a Drive:

<https://drive.google.com/file/d/1MNLatoVIUjbGQNteT0I3zd8WIML47KM/view?usp=sharing>

En esta, el robot parte del control manual con mando infrarrojos (distancia > 5 cm) y es dirigido a un depósito de arroz. Allí al detectar una distancia < 5 cm se detiene y al haber llegado en una posición bastante recta, se pulsa el botón 2 (carga de pala hacia el centro). Tras acabar el movimiento, se da para atrás y se retoma el control manual buscando el depósito de papel. A este se llega en una posición algo más desviada por lo que se pulsa el botón 6 (descarga de pala hacia la derecha). Al acabar el movimiento, se retrocede el robot hacia atrás recuperando de nuevo el control manual con mando.

12. Validación y conclusión

Tras terminar con el montaje y hacer uso de la programación llevada a cabo, se puede decir que se ha cumplido con los objetivos propuestos.

Se seleccionó una electrónica, se diseñó un robot que la ensamblase y supliera las funciones básicas de una excavadora, se exportó a V-REP CoppeliaSim, se programaron sus componentes con Arduino IDE y se realizó el montaje y recreación de una escena programada.

Sin embargo, hay detalles que no se tuvieron en cuenta o no pudieron predecirse y se tratan a continuación.

Respecto al montaje, no se esperó que las pilas aumentasen el grosor del conjunto de soporte y pilas de forma considerable. Por suerte no fue un problema que llevase a un cambio en el diseño y se pudo ensamblar igualmente gracias a los pequeños espacios milimétricos entre componentes que se habían dejado a modo de seguridad por detalles como este.

Respecto al material en la fabricación, se comenzó a fabricar en la universidad y por motivos ajenos a este proyecto, no pudo seguirse fabricando allí. Con el material y la fabricación de la universidad, por ejemplo, una pieza con un saliente para encaje de 30 mm encajaba sin necesidad de tornillos en su correspondiente encaje, de 30 mm también, sin holguras. Sin embargo, al realizarse el resto de las piezas de corte láser del robot en una empresa a parte, ese mismo encaje quedaba con holgura necesitando de tornillos que los fijasen correctamente. No se tenían tornillos en todos los encajes del diseño, por lo que supuso un problema, sobre todo en la parte de la pala. Para fijar las piezas en esta y en el resto de las partes que así lo necesitasen, se tuvo que utilizar pegamento líquido.

Respecto al vídeo realizado de recreación de una escena, se observa que la pala no se carga como cabría esperar. Esto es debido a que la pala no tiene una estructura "afilada" que penetre adecuadamente en el arroz utilizado para simular arena. Además, si a esto se le suma que los servomotores SG90 no son de gran potencia, se tiene el resultado obtenido. Quizá con arena real se hubieran tenido mejores resultados que con arroz. Sin embargo, el montaje se realizó en el último momento debido a los problemas surgidos de no poder realizar la fabricación de piezas en la

escuela tal y como se contaba y, en consecuencia, se realizó la escena con los materiales disponibles.

Respecto a la exportación y preparación para la simulación en CoppeliaSim, hubiera sido interesante, con más tiempo, realizar una simulación de una escena y compararla con la realidad, de forma que se puedan corregir errores de un modo más práctico y visual que utilizando scripts de Matlab para la comprobación de cálculos como se ha hecho en este proyecto.

Respecto a la gran cantidad de cables utilizados, son mayormente debidos a las conexiones de los motores DC con el circuito integrado L293D, que sumadas a las de los servomotores se obtiene el resultado de la ilustración 73. Una mejora para esto sería la realización de una PCB personalizada, o en su defecto, utilizar un módulo distinto a la placa de expansión Protoshield para Arduino UNO. En ese caso, habría que pensar si todos los componentes utilizados pudieran conectarse, que es la razón por la que se utilizó esta placa.

Como principal mejora, se propone el control y programación con otro tipo de tarjeta distinta a Arduino, como puede ser Raspberry Pi, aprovechando que se tiene una tarjeta ESP32-CAM la cual puede servir para múltiples proyectos basados por ejemplo en visión artificial.

13. Bibliografía

Arduino. ¿Qué es Arduino? [En línea] <https://arduino.cl/que-es-arduino/>.

Electronilab. Motorreductor con caja reductora 6V 1:48. [En línea] <https://electronilab.co/tienda/motorreductor-con-caja-reductora-6v-1-48/>.

Gaona, Diego. Tutorial Curso Básico en Solidworks Principiantes. [En línea] https://www.youtube.com/watch?v=dZrup1VqtVI&list=PL8zTQ91YahgXx-85IWgp66_V2pQJI39EL.

Hisour. Manipuladores móviles. [En línea] <https://www.hisour.com/es/mobile-manipulator-42888/>.

Naylamp Mechatronics. SENSOR ULTRASONIDO HC-SR04. [En línea] <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>.

Open Circuit. ESP32-CAM módulo de cámara basado en ESP32. [En línea] <https://opencircuit.es/Producto/Waveshare-ESP32-CAM-m%C3%B3dulo-c%C3%A1mara-basado>.

Piensa3D. Cómo mover motor DC en 2 direcciones con L293D. [En línea] <https://piensa3d.com/tutorial-arduino-mover-motor-dos-direcciones-l293d/>.

Poole, Gregory. Guía de los diferentes tipos y tamaños de excavadoras. [En línea] <https://www.gregorypoole.com/es/guide-to-the-different-types-and-sizes-of-excavators/>.

Random Nerd Tutorials. ESP32-CAM Video Streaming and Face Recognition with Arduino IDE. [En línea] <https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/>.

Spanish Alibaba. 4,8 V- 6V SG90 Micro servo motor 9g para robots. [En línea] <https://spanish.alibaba.com/product-detail/4-8v-6v-sg90-micro-servo-motor-9g-for-robots-60781304037.html>.

Tech StudyCell. How to program ESP32-CAM using Arduino UNO. [En línea] <https://www.youtube.com/watch?v=q-KlpFibRMk>.

Trotec. Madera para corte y grabado láser . [En línea] <https://www.troteclaser.com/es/noticias-laser/madera-trotec-para-corte-y-grabado-laser/> .

Unit Electronics. Motorreductor amarillo. [En línea] <https://uelectronics.com/producto/motorreductor-amarillo-para-carrito/>.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Anexo I:
Robot meArm

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

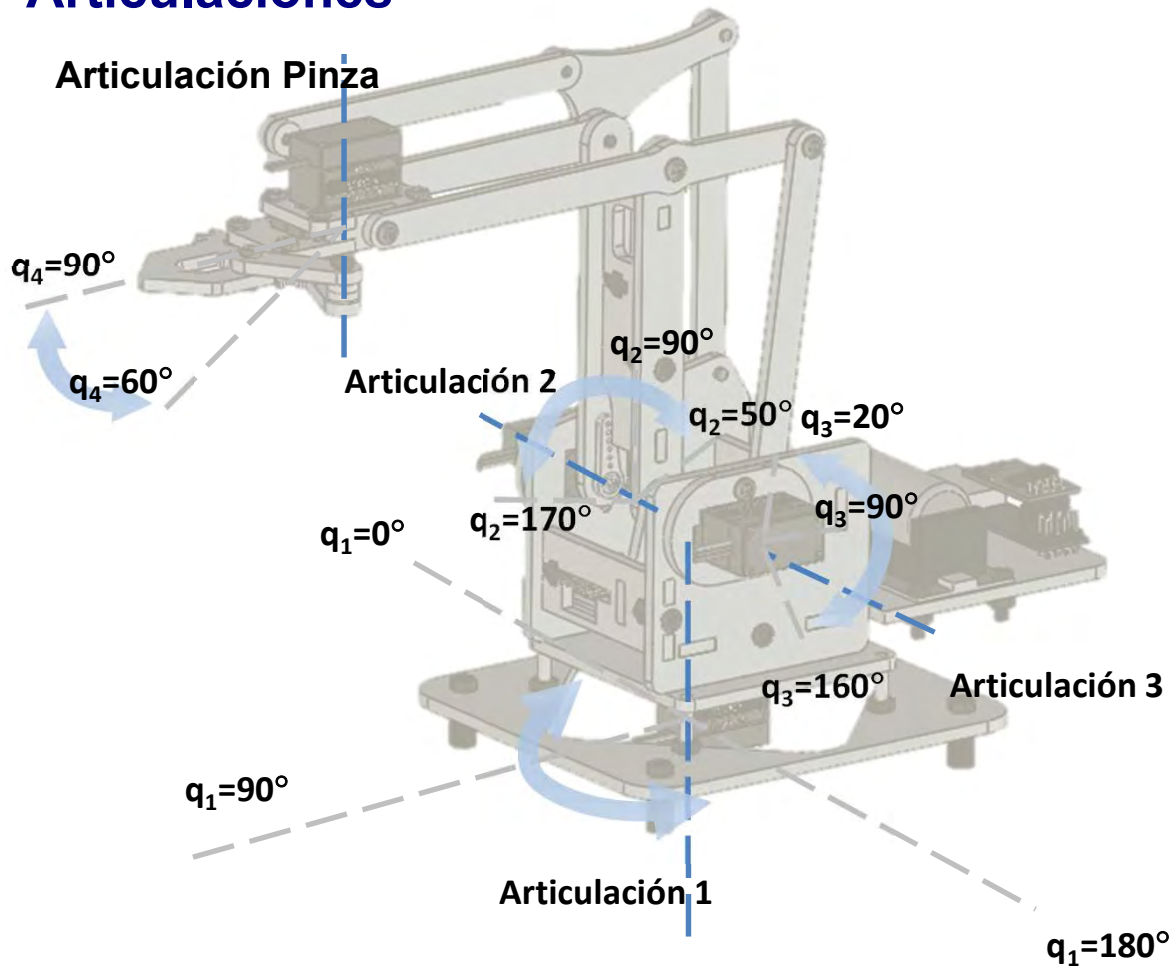
David Martínez Esteso

TUTORIZADO POR

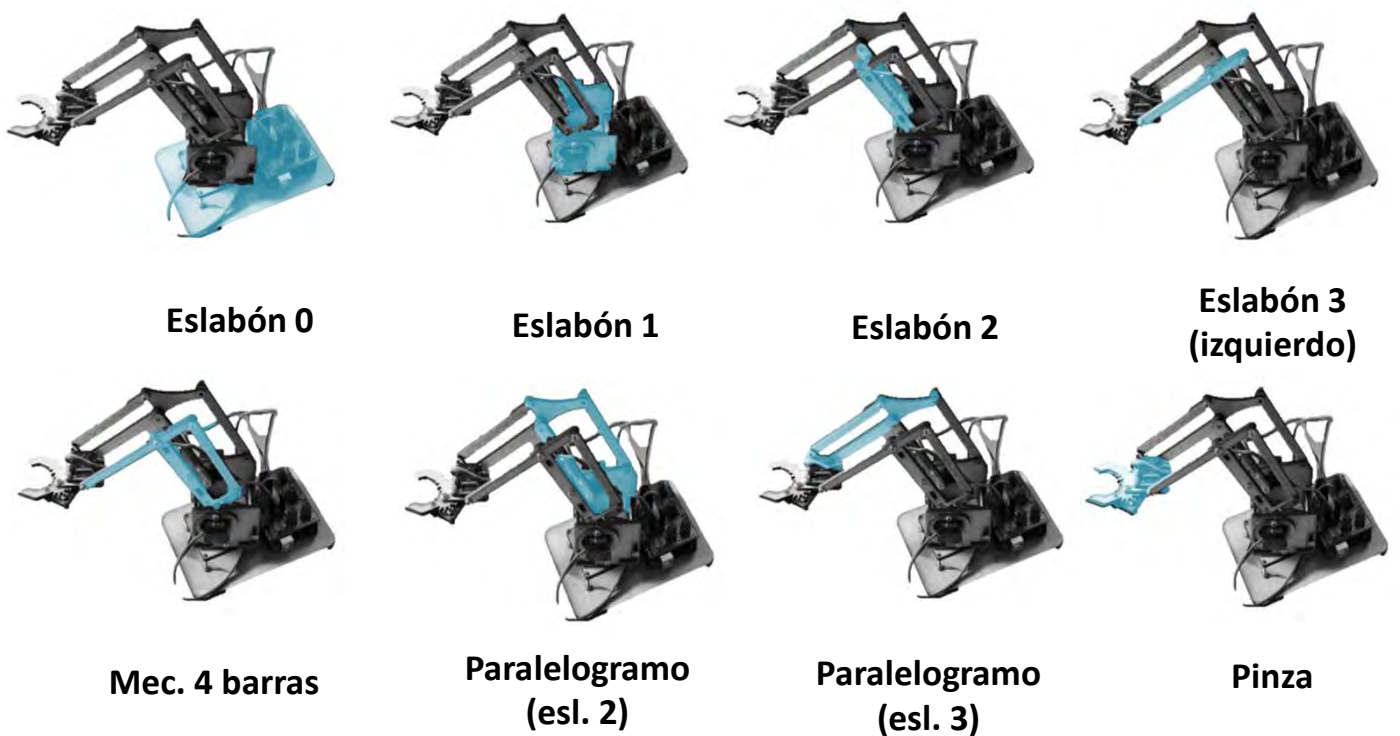
Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

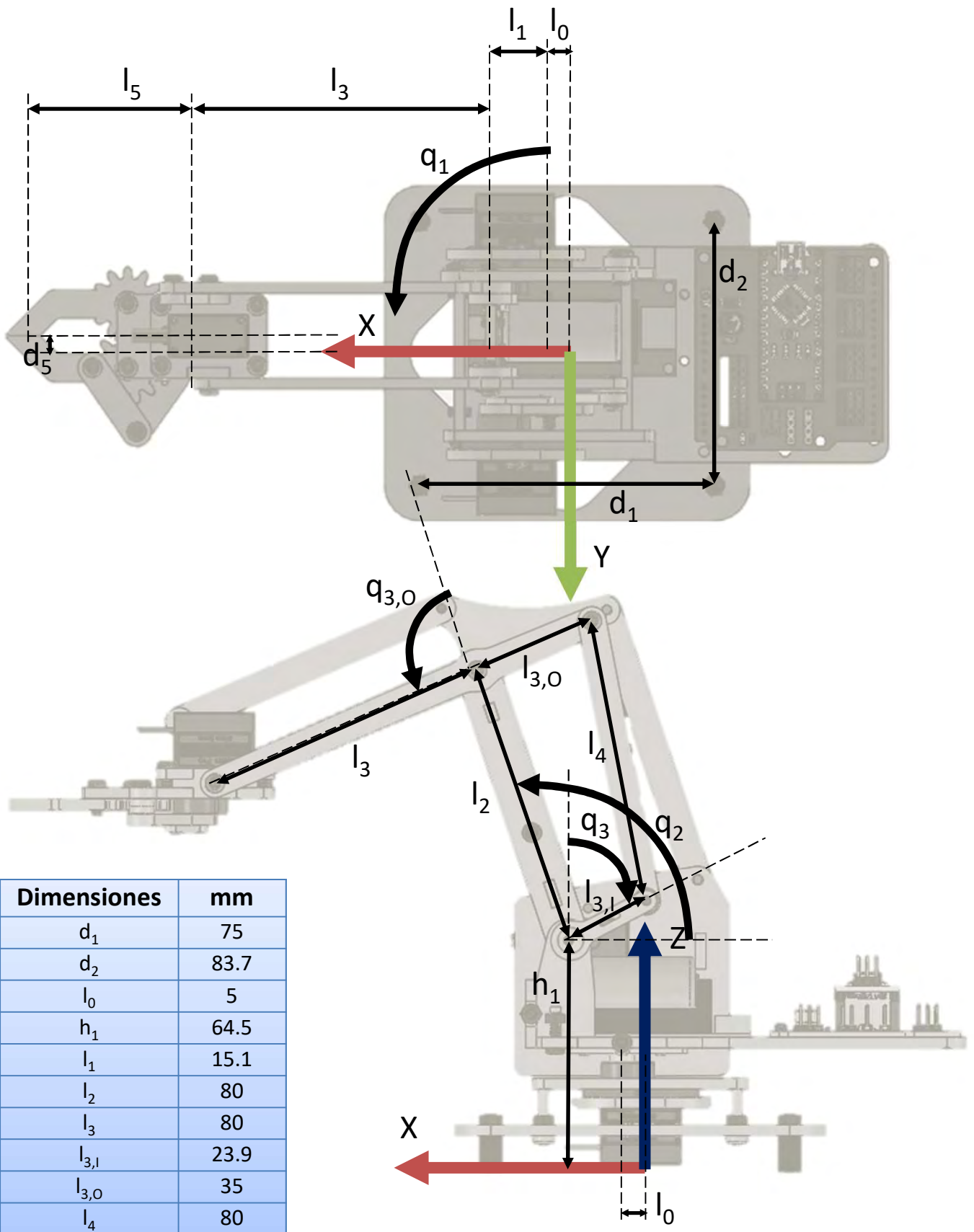
Articulaciones



Eslabones

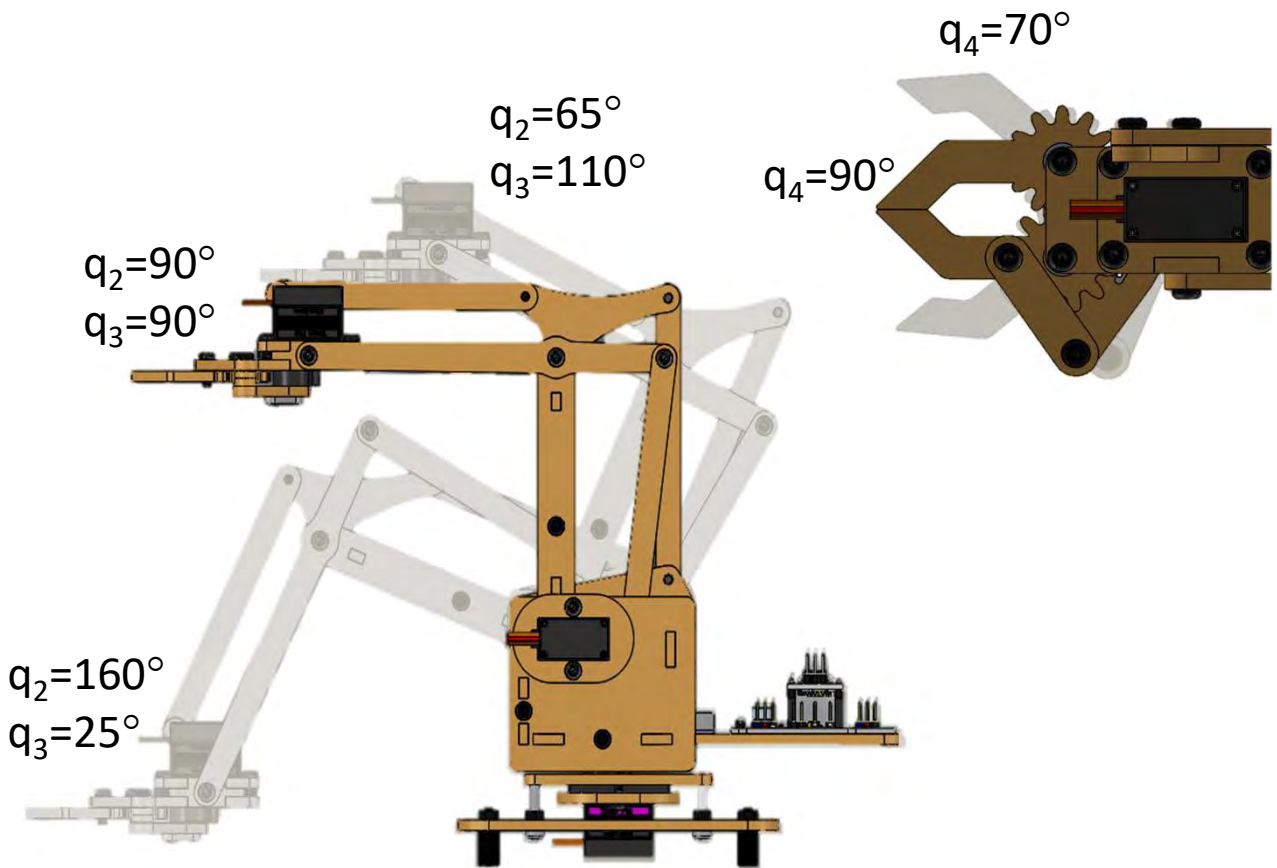
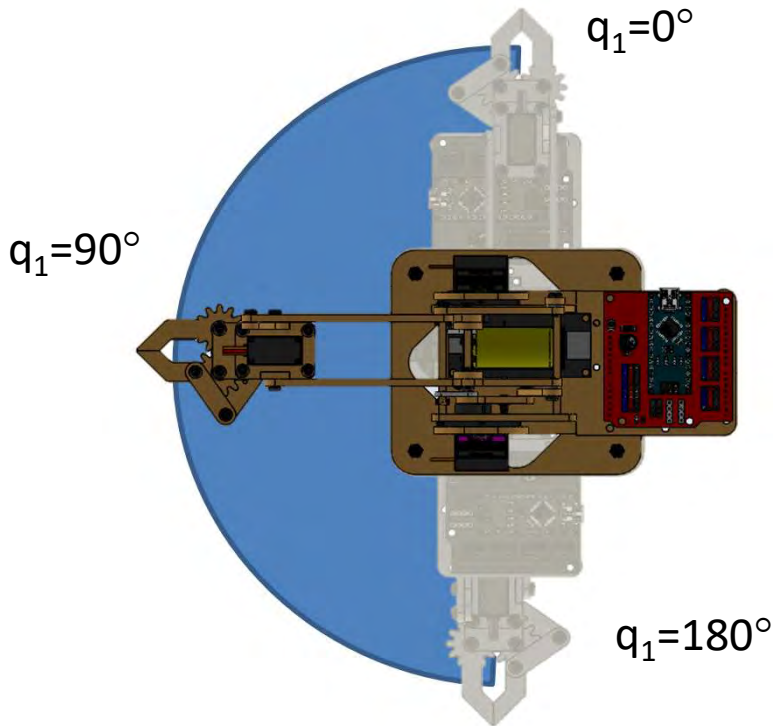


Dimensiones del robot



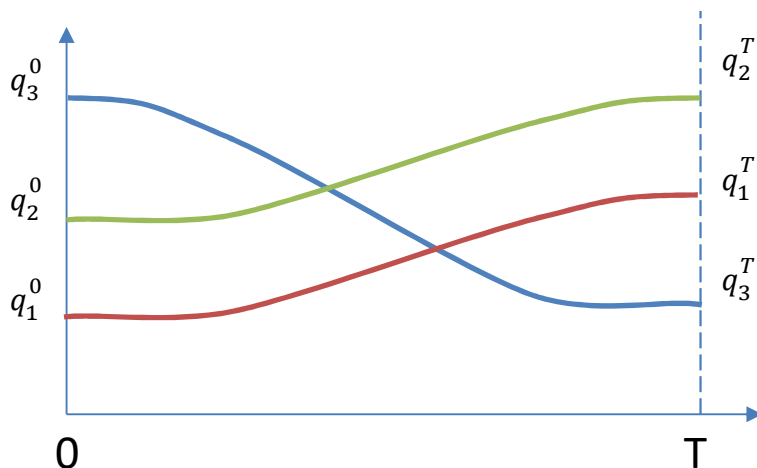
Dimensiones	mm
d_1	75
d_2	83.7
l_0	5
h_1	64.5
l_1	15.1
l_2	80
l_3	80
$l_{3,1}$	23.9
$l_{3,0}$	35
l_4	80
l_5	52
d_5	5

Movimientos del robot



Control eje a eje con trayectoria polinomial

En el control eje a eje, el robot debe moverse a una configuración articular determinada por los valores de referencia en un tiempo dado. En este modo, los ejes están coordinados, en el sentido de que cada uno de ellos trata de alcanzar una posición destino, pero todos al mismo tiempo.



Se define una trayectoria suave, $q_i(t)$, en el espacio articular mediante una expresión polinomial (la derivada de la trayectoria es $\dot{q}_i(t)$) y se establecen unas condiciones de contorno, $q_i(0)$ y $q_i(T)$ y sus derivadas, $\dot{q}_i(0)$ y $\dot{q}_i(T)$. En concreto, para una trayectoria cúbica, podemos establecer que las condiciones de contorno de la trayectoria serán el punto inicial q_i^0 y final q_i^T a alcanzar, respectivamente; y que las velocidades iniciales y finales serán cero. La velocidad de la trayectoria se obtiene a partir de la derivada de la misma. Con todo esto, se una expresión cúbica de la trayectoria como se muestra, siendo:

Trayectoria (posición): $q_i(t) = a_i t^3 + b_i t^2 + c_i t + d_i$

Trayectoria (velocidad): $\dot{q}_i(t) = 3a_i t^2 + 2b_i t + c_i$

Condiciones de contorno: $q_i(0) = q_i^0$ $q_i(T) = q_i^T$
 $\dot{q}_i(0) = 0$ $\dot{q}_i(T) = 0$

Con las condiciones de contorno, podemos obtener los valores de los parámetros de la trayectoria cúbica:

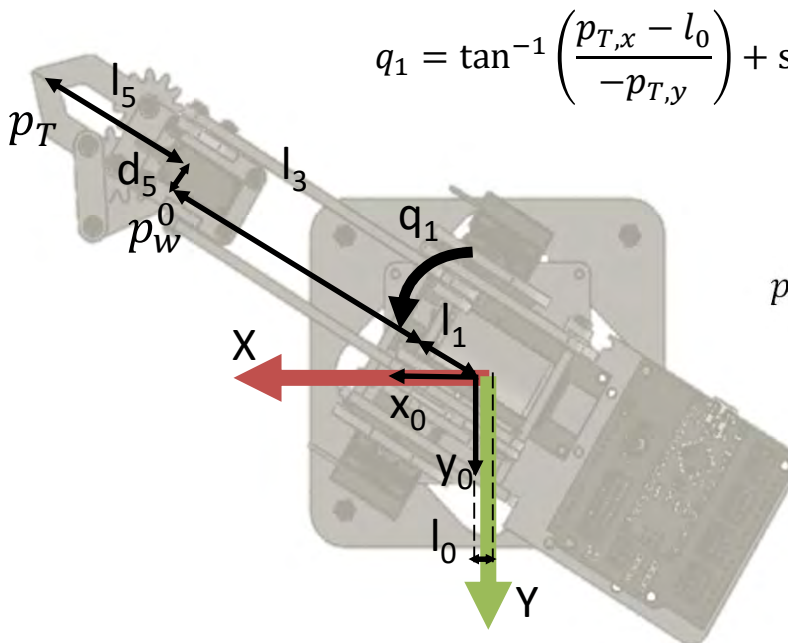
$$a_i = -\frac{2(q_i^T - q_i^0)}{T^3} \quad b_i = \frac{3(q_i^T - q_i^0)}{T^2} \quad c_i = 0 \quad d_i = q_i^0$$

Los parámetros de la trayectoria son diferentes para cada servo y debe recalcularse cada vez que se modifique la posición inicial, final, velocidades o el tiempo de la trayectoria.

Cinemática inversa

Si la información de entrada es dada como una posición en el espacio Cartesiano, debemos ahora calcular la posición final de las articulaciones a partir de la cinemática inversa del robot. El cálculo se realizará en varios pasos:

1) El ángulo q_1 se calcula a partir de las coordenadas x e y del punto de la pinza p_T . Debido a que el término d_5 no es cero, el cálculo de q_1 se debe realizar mediante la suma de dos ángulos: el primero corresponde al ángulo de p_T con respecto al sistema de referencia 0 (que se muestra en la figura). Luego, debemos compensar el ángulo debido a la distancia d_5 que se obtiene de una relación trigonométrica sencilla. Por otro lado, el punto de la base de la pinza del robot, p_w^0 , con respecto al sistema de referencia del eslabón 1 centrado sobre la articulación 1 se calcula a partir del punto objetivo de la pinza, p_T , como



$$q_1 = \tan^{-1} \left(\frac{p_{T,x} - l_0}{-p_{T,y}} \right) + \sin^{-1} \left(\frac{d_5}{\sqrt{(p_{T,x} - l_0)^2 + p_{T,y}^2}} \right)$$

$$p_w^0 = p_T + \begin{bmatrix} -l_5 \sin q_1 - l_0 \\ l_5 \cos q_1 \\ 0 \end{bmatrix}$$

2) Las coordenadas q_2 y $q_{3,0}$ se calculan a partir del triángulo formado por los eslabones 2 y 3:

$$r = \sqrt{p_{w,x}^2 + p_{w,y}^2} - l_1$$

$$z_e = p_{w,z} - h_1$$

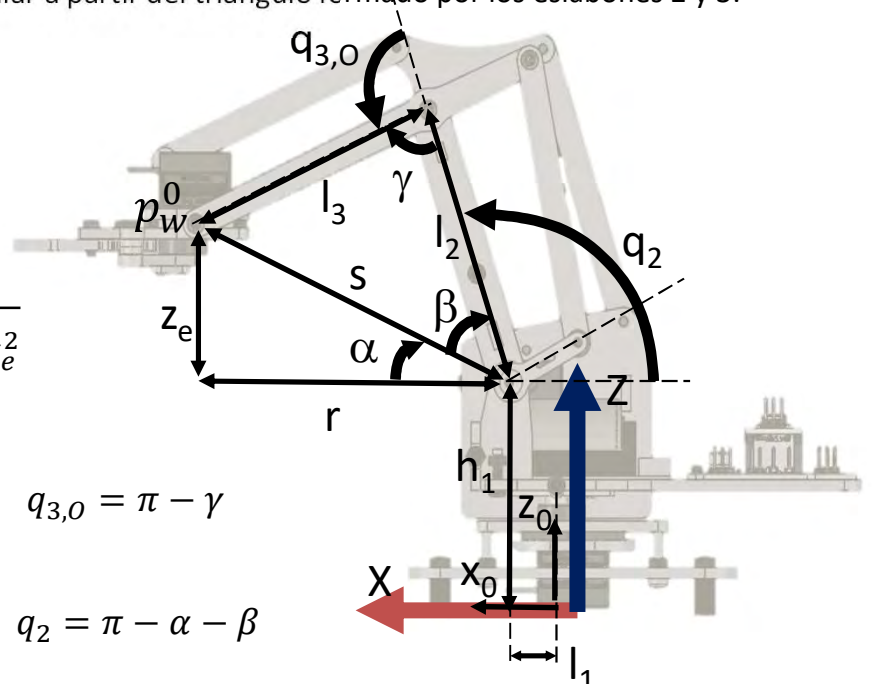
$$\alpha = \tan^{-1} \frac{z_e}{r} \quad s = \sqrt{r^2 + z_e^2}$$

$$\gamma = \cos^{-1} \left(\frac{l_2^2 + l_3^2 - s^2}{2l_2l_3} \right)$$

$$\beta = \cos^{-1} \left(\frac{s^2 + l_2^2 - l_3^2}{2sl_2} \right)$$

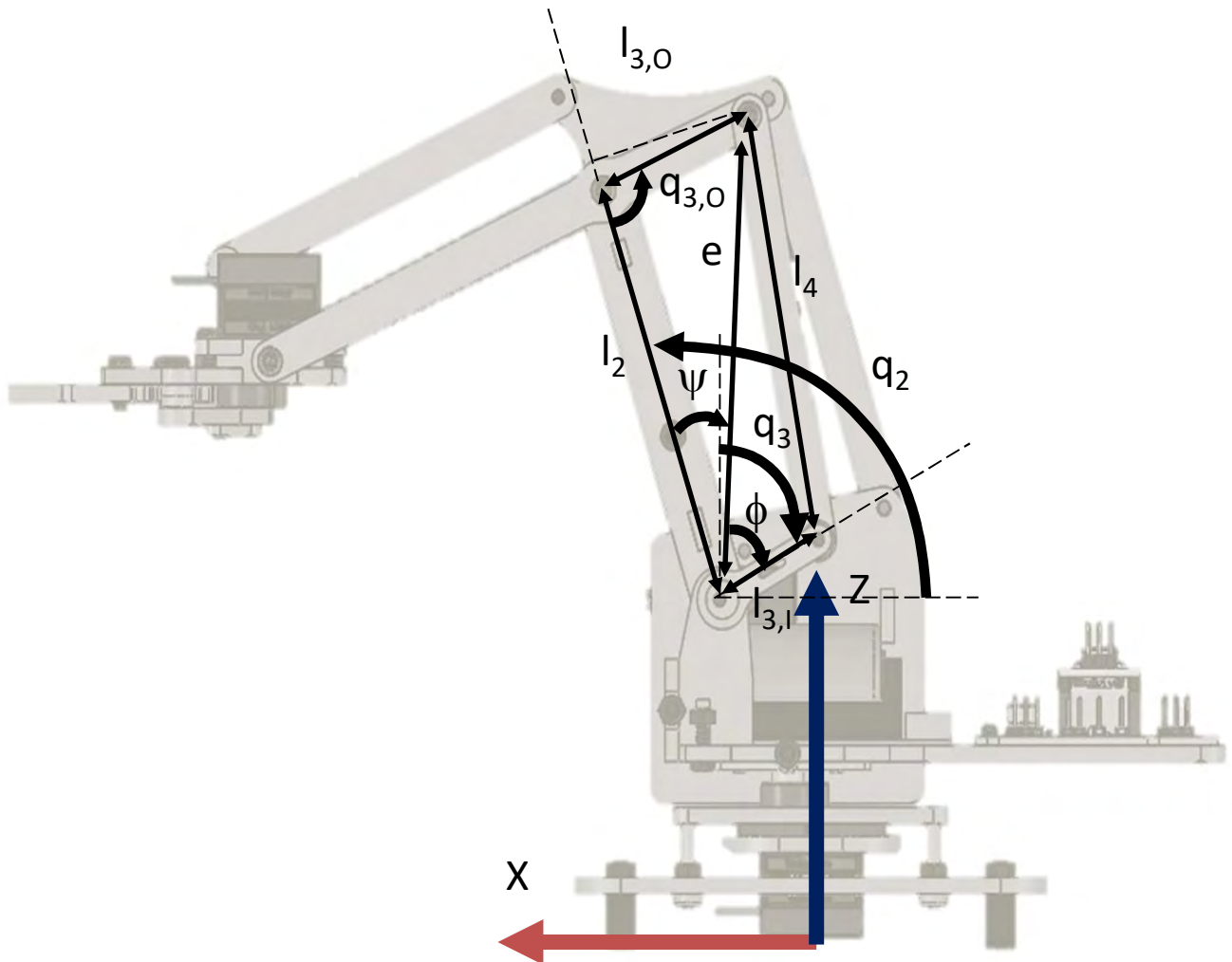
$$q_{3,0} = \pi - \gamma$$

$$q_2 = \pi - \alpha - \beta$$



Mecanismo de 4 barras ($q_{30} \rightarrow q_3$)

3) Finalmente, el ángulo de la tercera articulación q_3 se calcula a partir de la cinemática del mecanismo de 4 barras:



$$e = \sqrt{l_{3,0}^2 + l_2^2 - 2l_{3,0}l_2 \cos q_{3,0}}$$

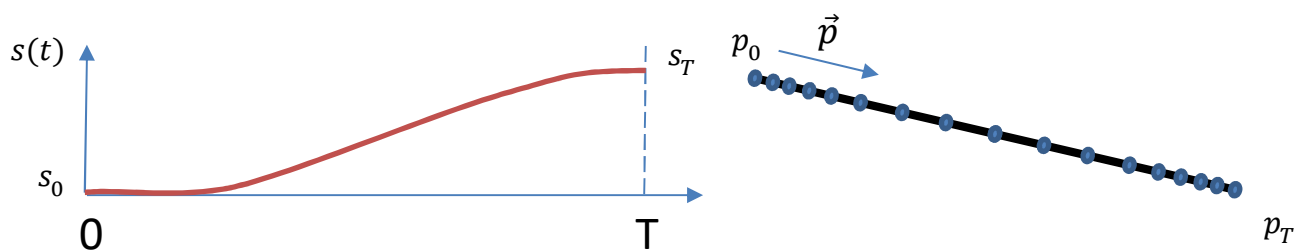
$$\psi = \sin^{-1} \left(\frac{l_{3,0} \sin q_{3,0}}{e} \right)$$

$$\phi = \cos^{-1} \left(\frac{e^2 + l_{3,1}^2 - l_4^2}{2el_{3,1}} \right)$$

$$q_3 = \psi + \phi + \frac{\pi}{2} - q_2$$

Control lineal con trayectoria polinomial

El control lineal de movimientos con trayectoria polinomial se plantea como un problema en el que el avance lineal $s(t)$ se implementa con una trayectoria polinomial, mientras que la dirección de avance se determina por un vector unitario \vec{p} a partir de las posiciones de inicio p_0 y destino p_T . De esta forma, el avance de la trayectoria tendrá una velocidad nula al inicio y al final de la misma, mientras que en los puntos intermedios, la velocidad será mayor (esto se refleja en la siguiente figura).



Por tanto, la trayectoria lineal se define como: $p(t) = s(t)\vec{p} + p_0$

$$s(t) = at^3 + bt^2 + ct + d \quad s(0) = 0 \quad \dot{s}(0) = 0$$

$$\dot{s}(t) = 3at^2 + 2bt + c \quad s(T) = \|p_T - p_0\|_2 \quad \dot{s}(T) = 0$$

con

$$a = -\frac{2\|p_T - p_0\|_2}{T^3} \quad b = \frac{3\|p_T - p_0\|_2}{T^2} \quad c = 0 \quad d = 0 \quad \vec{p} = \frac{p_T - p_0}{\|p_T - p_0\|_2}$$

Siendo $\|\cdot\|_2$ el operador para el cálculo de la distancia Euclídea, es decir:

$$\|p_T - p_0\|_2 = \sqrt{(p_{T,x} - p_{0,x})^2 + (p_{T,y} - p_{0,y})^2 + (p_{T,z} - p_{0,z})^2}$$

Cinemática directa

Para la implementación del control lineal, será necesario calcular la posición de la pinza a partir de la configuración actual, es decir, los ángulos de los servos. Al igual que anteriormente, se procede a explicar los cálculos necesarios para implementar la cinemática directa del robot.

La separación de la muñeca con respecto al eje de la articulación 1 se calcula como (en la figura, cuando $q_2 = \frac{\pi}{2}$ y $q_{3,o} = \frac{\pi}{2}$ corresponde a la distancia $l_1 + l_3$):

$$r = l_1 - l_2 \cos q_2 - l_3 \cos(q_2 + q_{3,o})$$

Por otro lado, la altura de la pinza se calcula a partir de:

$$p_{T,z} = h_1 + l_2 \sin q_2 + l_3 \sin(q_2 + q_{3,o})$$

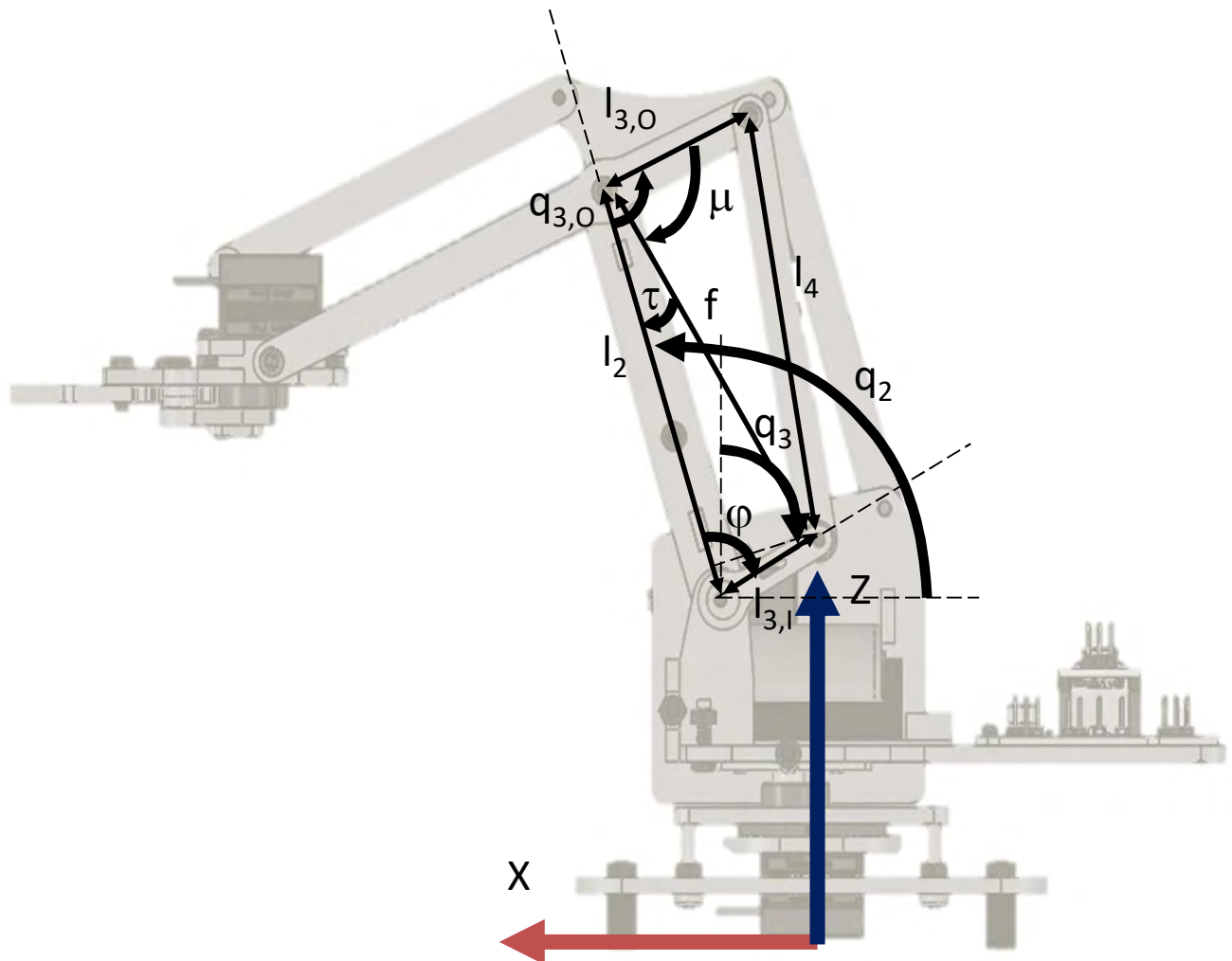
Mientras que las coordenadas x e y de la punta de la pinza son:

$$p_{T,x} = l_0 + (r + l_5) \sin q_1 - d_5 \cos q_1$$

$$p_{T,y} = -(r + l_5) \cos q_1 - d_5 \sin q_1$$

Mecanismo de 4 barras ($q_3 \rightarrow q_{30}$)

El ángulo del eslabón 3, q_3 , se calcula a partir de la cinemática del mecanismo de 4 barras:



$$\varphi = q_3 + q_2 - \frac{\pi}{2}$$

$$f = \sqrt{l_{3,I}^2 + l_2^2 - 2l_{3,I}l_2 \cos \varphi}$$

$$\mu = \cos^{-1} \left(\frac{f^2 + l_{3,0}^2 - l_4^2}{2fl_{3,0}} \right)$$

$$\tau = \sin^{-1} \left(\frac{l_{3,I} \sin \varphi}{f} \right)$$

$$q_{3,0} = \tau + \mu$$



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Anexo II:
Código de
programación
en Arduino IDE

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Esteso

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

```

//DC Motors

#define ENABLE1 6
#define IN1 2
#define IN2 3

#define ENABLE2 5
#define IN3 4
#define IN4 13

//Robotic arm

#include <Math.h>
#include <Servo.h>

Servo servos[12]; //Max number of digital signals

#define Pi 3.141592653589793
#define JOINTS 3

typedef struct
{
    uint8_t pin;
    int offset;
    int min_pos;
    int max_pos;
} RobotServo_t;
RobotServo_t robotServos[JOINTS]={{8,-5, 65,115},{9,-5,90, 175},{10,7,0,95}};

typedef struct
{
    double l0;
    double h1;
    double l1;
    double l2;
    double l3;
    double l3l;
    double l3O;
    double l4;
    double l5;
    double d5;
} RobotParams_t;
RobotParams_t params={5.0,87.5,15.1,80.0,80.0,23.9,35.0,80.0,10.7,0.0};

typedef struct
{
    double x;
    double y;
    double z;
} RobotPosition_t;
RobotPosition_t pT0={130.24,-10.7,118.32};
RobotPosition_t pT1={160,-10.7,53};
RobotPosition_t pT2={135, -10.7, 53};
RobotPosition_t pT3={143.4, 83.82, 53};
RobotPosition_t pT4={123.1, 69.64, 53};
RobotPosition_t pT5={146.78, -77.92, 53};
RobotPosition_t pT6={124.36, -67.47, 53};

```

```
double q0[JOINTS]={90.0,120.0,60.0};//servos initial position in angles
double qX[JOINTS]; //qX keeps the last position in angles after each commmand moveJ, which is the initial of
the next movement.
```

```
//IR Sensor
```

```
#include <IRremote.h>
```

```
#define IR_PIN A0
```

```
#define CODE_ON 0xA25D // on/off
```

```
#define CODE_UP 0x906F // up
```

```
#define CODE_DOWN 0xE01F // down
```

```
#define CODE_LEFT 0x22DD // left
```

```
#define CODE_RIGHT 0xC23D // right
```

```
#define CODE_STOP 0x02FD // Stop Button
```

```
#define CODE_0 0x6897 // 0
```

```
#define CODE_1 0x30CF // 1
```

```
#define CODE_2 0x18E7 // 2
```

```
#define CODE_3 0x7A85 // 3
```

```
#define CODE_4 0x10EF // 4
```

```
#define CODE_5 0x38C7 // 5
```

```
#define CODE_6 0x5AA5 // 6
```

```
//#define CODE_7 0xFF42BD // 7
```

```
//#define CODE_8 0xFF4AB5 // 8
```

```
//#define CODE_9 0xFF52AD // 9
```

```
//#define CODE_REP 0xFFFFFFFF // key repeat
```

```
IRrecv receiver(IR_PIN); // IR Sensor
```

```
bool is_running_forward = false;
```

```
bool is_running_backward = false;
```

```
bool is_running_right = false;
```

```
bool is_running_left = false;
```

```
unsigned int readIR() {
```

```
  decode_results results;
```

```
  if (receiver.decode(&results)) {
```

```
    receiver.resume();
```

```
    return results.value;
```

```
  }
```

```
  return 0;
```

```
}
```

```
float distance=0;
```

```
void setup()
```

```
{
```

```
  //DC Motors & HC-SR04
```

```
  pinMode(13,INPUT);
```

```
  pinMode(ENABLE1, OUTPUT);
```

```
  pinMode(IN1, OUTPUT);
```

```
  pinMode(IN2, OUTPUT);
```

```
  pinMode(ENABLE2, OUTPUT);
```

```
  pinMode(IN3, OUTPUT);
```

```
  pinMode(IN4, OUTPUT);
```

```

//IR Sensor

pinMode(IR_PIN,INPUT);
receiver.enableIRIn();

//Serial

Serial.begin(9600);

//Sets the robot to the initial position
for (int i=0;i<JOINTS;i++)
  writeServo(robotServos[i],(int)q0[i]);
  delay(3000);

//Shovel to close position
servos[11].attach(11);
servos[11].write(50);
}

//Wheel functions

void stopLeftWheel(void){
  analogWrite(ENABLE1, 0);
}

void stopRightWheel(void){
  analogWrite(ENABLE2, 0);
}

void rightWheelForward(void){
  analogWrite(ENABLE1, 255);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
}

void rightWheelBackward(void){
  analogWrite(ENABLE1, 255);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
}

void leftWheelForward(void){
  analogWrite(ENABLE2, 255);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
}

void leftWheelBackward(void){
  analogWrite(ENABLE2, 255);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
}

//distance measure HC-SR04 ultrasonic sensor

```

```

float distanceMeasure(int triggerpin, int echopin){

    pinMode(triggerpin,OUTPUT);
    digitalWrite(triggerpin,LOW);
    delayMicroseconds(2);
    digitalWrite(triggerpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerpin,LOW);
    pinMode(echopin,INPUT);

    return(pulseIn(echopin,HIGH)*0.01723);
}

void loop()
{
    unsigned int button = readIR();
    distance = distanceMeasure(12,7);
    Serial.println("Distancia:");
    Serial.println(distance);
    Serial.println("boton:");
    Serial.println(button,HEX);
    if (distance>5){ //IR remote control

        if (button == CODE_UP) {
            is_running_forward = true;
            is_running_backward = false;
            is_running_left = false;
            is_running_right = false;
        }
        if (button == CODE_DOWN) {
            is_running_forward = false;
            is_running_backward = true;
            is_running_left = false;
            is_running_right = false;
        }
        if (button == CODE_LEFT) {
            is_running_forward = false;
            is_running_backward = false;
            is_running_left = true;
            is_running_right = false;
        }
        if (button == CODE_RIGHT) {
            is_running_forward = false;
            is_running_backward = false;
            is_running_left = false;
            is_running_right = true;
        }
        if (button == CODE_STOP) {
            is_running_forward = false;
            is_running_backward = false;
            is_running_left = false;
            is_running_right = false;
            stopLeftWheel();
            stopRightWheel();
        }

        if (is_running_forward) {
            // Code to move forward the robot

```

```

    leftWheelForward();
    rightWheelForward();
}
if (is_running_backward) {
    // Code to move backward the robot
    leftWheelBackward();
    rightWheelBackward();
}
if (is_running_left) {
    // Code to move left the robot
    leftWheelBackward();
    rightWheelForward();
}
if (is_running_right) {
    // Code to move right the robot
    leftWheelForward();
    rightWheelBackward();
}
}
else { //object detection, robotic arm movement

    is_running_forward = false;
    is_running_backward = false;
    is_running_left = false;
    is_running_right = false;

    stopLeftWheel();
    stopRightWheel();

    if (button == CODE_1){ //charge movement to the left
        shovelChargeMovement(pT3, pT4);
    }

    if (button == CODE_2){ //charge movement to the middle
        shovelChargeMovement(pT1, pT2);
    }

    if (button == CODE_3){ //charge movement to the right
        shovelChargeMovement(pT5, pT6);
    }

    if (button == CODE_4){ //discharge movement to the left
        shovelDischargeMovement(pT3);
    }

    if (button == CODE_5){ //discharge movement to the middle
        shovelDischargeMovement(pT1);
    }

    if (button == CODE_6){ //discharge movement to the left
        shovelDischargeMovement(pT5);
    }

    if (button == CODE_DOWN){ //go back
        double InTime = millis();
        while (millis())<InTime+1500){
            leftWheelBackward();
            rightWheelBackward();

```



```

    }
    stopLeftWheel();
    stopRightWheel();
  }
}

void shovelChargeMovement (const RobotPosition_t target1, const RobotPosition_t target2){

  moveJ(robotServos,q0,target1,3,params); //movement from pT0 to target1
  delay(200);

  for (int pos=50; pos<125; pos++){ //open shovel
    servos[11].attach(11);
    servos[11].write(pos);
    delay(20);
  }

  moveJ(robotServos,qX,target2,3,params); //movement from last position (target1) to target2

  for (int pos=125; pos>50; pos--){ //close shovel
    servos[11].attach(11);
    servos[11].write(pos);
    delay(20);
  }

  moveJ(robotServos,qX,pT0,3,params); //movement from last position (target2) to pT0

}

void shovelDischargeMovement (const RobotPosition_t target){

  moveJ(robotServos,q0,target,3,params); //movement from pT0 to target
  delay(200);

  for (int pos=50; pos<125; pos++){ //open shovel
    servos[11].attach(11);
    servos[11].write(pos);
    delay(20);
  }

  delay(1000);

  for (int pos=125; pos>50; pos--){ //close shovel
    servos[11].attach(11);
    servos[11].write(pos);
    delay(20);
  }

  moveJ(robotServos,qX,pT0,3,params); //movement from last position (target) to pT0

}

void writeServo(const RobotServo_t &servo, int angle)
{
  angle=constrain(angle+servo.offset,servo.min_pos,servo.max_pos);

  servos[servo.pin].attach(servo.pin);

```

```

servos[servo.pin].write(angle);

Serial.print("S");
Serial.print(servo.pin);
Serial.print(":");
Serial.print((int)angle,DEC);
Serial.println(";");
}

//Joint movement function
void moveAbsJ(const RobotServo_t robotServos[JOINTS], const double qN[JOINTS], const double qT[JOINTS],
const double T)
{
double a[JOINTS],b[JOINTS],c[JOINTS],d[JOINTS],q,t,t0;
//TODO: Compute trajectory parameters for each joint

for(int i=0;i<JOINTS;i++)
{
a[i]=(-2*(qT[i]-qN[i])/pow(T,3));
b[i]=(3*(qT[i]-qN[i])/pow(T,2));
c[i]=0.0;
d[i]=qN[i];
}

t0=millis()/1000.0; //initial time
t=0.0; //time count
while(t<T)
{
for (int i=0;i<JOINTS;i++)
{
//TODO: Evaluate the trajectory for joint "i" at time "t" and store the result in "q".
q=a[i]*pow(t,3)+b[i]*pow(t,2)+c[i]*t+d[i];
writeServo(robotServos[i],(int)q);
}
delay(20);
t=millis()/1000.0-t0;
}
}

//Point to point movement function
void moveJ(const RobotServo_t robotServos[JOINTS],const double qN[JOINTS], const RobotPosition_t
target,const float T, const RobotParams_t &params)
{
double qT[JOINTS];
inverseKin(target,params,qT);
moveAbsJ(robotServos,qN,qT,5);
qX[0]=qT[0];
qX[1]=qT[1];
qX[2]=qT[2];
Serial.println(qX[0]);
Serial.println(qX[1]);
Serial.println(qX[2]);
}

//Inverse kinematics
void inverseKin(const RobotPosition_t &target,const RobotParams_t &params, double *q)

```

```

{
q[0]=atan2(target.x-params.l0,-target.y)+asin(params.d5/sqrt(pow(target.x-params.l0,2)+pow(target.y,2)));

double pw[3][1]={{target.x-params.l5*sin(q[0])-params.l0},{target.y+params.l5*cos(q[0])},{target.z}};
double r=sqrt(pow(pw[0][0],2)+pow(pw[1][0],2))-params.l1;
double ze=(pw[2][0]-params.h1);
double alfa=atan2(ze,r);
double s=sqrt(pow(r,2)+pow(ze,2));
double gamma=acos(((pow(params.l2,2))+pow(params.l3,2))-(pow(s,2)))/(2*params.l2*params.l3));
double beta=acos(((pow(s,2))+pow(params.l2,2))-(pow(params.l3,2)))/(2*s*params.l2));

q[1]=Pi-alfa-beta;

double q30=Pi-gamma;
double e=sqrt((pow(params.l30,2))+pow(params.l2,2))-(2*params.l30*params.l2*cos(q30)));
double chi=asin((params.l30*sin(q30))/(e));
double fi=acos(((pow(e,2))+pow(params.l31,2))-(pow(params.l4,2)))/(2*e*params.l31));

q[2]= chi+fi+(Pi/2)-q[1];

q[0]=q[0]*180/Pi;
q[1]=q[1]*180/Pi;
q[2]=q[2]*180/Pi;
}

```



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Anexo III:
Código de
programación
en MATLAB

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Esteso

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

I. Cinemática Inversa:

```
%Introducir coordenadas cartesianas
x=135
y=-10.7
z=53

l0=5;
h1=87.5;
l1=15.1;
l2=80;
l3=80;
l3I=23.9;
l30=35;
l4=80;
d5=10.7;
l5=0;

q1=atan2(x-l0,-y)+asin(d5/sqrt((x-l0)^2+y^2));
pw=[ [x+d5*cos(q1)]; [y-d5*sin(q1)]; [z] ];

r=sqrt((pw(1))^2+(pw(2))^2)-l1;
ze=(pw(3)-h1);
alfa=atan2(ze,r);
s=sqrt(r^2+ze^2);
gamma=acos((l2^2+l3^2-s^2)/(2*l2*l3));
beta=acos((s^2+l2^2-l3^2)/(2*s*l2));
q2=pi-alfa-beta;
q30=pi-gamma;

e=sqrt(l30^2+l2^2-2*l30*l2*cos(q30));
chi=asin((l30*sin(q30))/e);
fi=acos((e^2+l3I^2-l4^2)/(2*e*l3I));
q3=chi+fi+(pi/2)-q2;

%Resultado en ángulos de los servomotores:
q1=q1*180/pi
q2=q2*180/pi
q3=q3*180/pi
```

II. Cinemática Directa:

```
%Introducir ángulos de los servomotores:
q1=90
q2=120
q3=60

l0=5;
h1=87.5;
l1=15.1;
l2=80;
l3=80;
l3I=23.9;
l30=35;
l4=80;
d5=10.7;
l5=0;

fi=q3*pi/180+q2*pi/180-pi/2;
```

```

f=sqrt((l3I^2)+(l2^2)-2*l3I*l2*cos(fi));
mu=acos(((f^2)+(l30^2)-(l4^2))/(2*f*l30));
tau=asin(l3I*sin(fi)/f);
q30=tau+mu;

r=l1-l2*cos(q2*pi/180)-l3*cos(q2*pi/180+q30);

%Resultado en coordenadas cartesianas:
ptx=l0+(r+l5)*sin(q1*pi/180)-d5*cos(q1*pi/180)
pty=-(r+l5)*cos(q1*pi/180)-d5*sin(q1*pi/180)
ptz=h1+l2*sin(q2*pi/180)+l3*sin(q2*pi/180+q30)

```



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Anexo IV:
Datasheet del circuito
integrado L293D

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Estesó

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

L293x Quadruple Half-H Drivers

1 Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

2 Applications

- Stepper Motor Drivers
- DC Motor Drivers
- Latching Relay Drivers

3 Description

The L293 and L293D devices are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.

The L293 and L293D are characterized for operation from 0°C to 70°C.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
L293NE	PDIP (16)	19.80 mm x 6.35 mm
L293DNE	PDIP (16)	19.80 mm x 6.35 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Logic Diagram

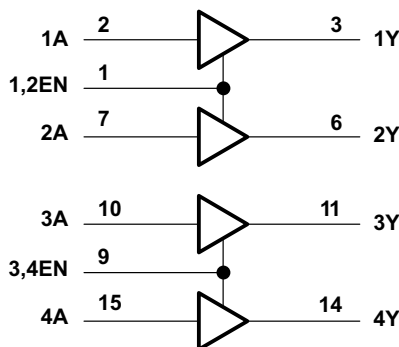


Table of Contents

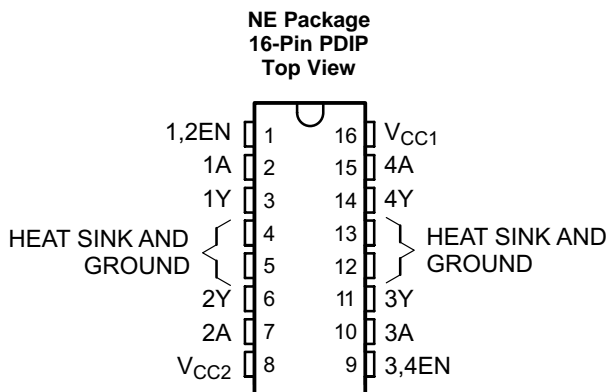
1 Features	1	8.3 Feature Description	7
2 Applications	1	8.4 Device Functional Modes	8
3 Description	1	9 Application and Implementation	9
4 Revision History	2	9.1 Application Information	9
5 Pin Configuration and Functions	3	9.2 Typical Application	9
6 Specifications	4	9.3 System Examples	10
6.1 Absolute Maximum Ratings	4	10 Power Supply Recommendations	13
6.2 ESD Ratings	4	11 Layout	14
6.3 Recommended Operating Conditions	4	11.1 Layout Guidelines	14
6.4 Thermal Information	4	11.2 Layout Example	14
6.5 Electrical Characteristics	5	12 Device and Documentation Support	15
6.6 Switching Characteristics	5	12.1 Related Links	15
6.7 Typical Characteristics	5	12.2 Community Resources	15
7 Parameter Measurement Information	6	12.3 Trademarks	15
8 Detailed Description	7	12.4 Electrostatic Discharge Caution	15
8.1 Overview	7	12.5 Glossary	15
8.2 Functional Block Diagram	7	13 Mechanical, Packaging, and Orderable Information	15

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision C (November 2004) to Revision D	Page
• Removed <i>Ordering Information</i> table	1
• Added <i>ESD Ratings</i> and <i>Thermal Information</i> tables, <i>Feature Description</i> section, <i>Device Functional Modes</i> , <i>Application and Implementation</i> section, <i>Power Supply Recommendations</i> section, <i>Layout</i> section, <i>Device and Documentation Support</i> section, and <i>Mechanical, Packaging, and Orderable Information</i> section.	1

5 Pin Configuration and Functions



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V _{CC1}	16	—	5-V supply for internal logic translation
V _{CC2}	8	—	Power VCC for drivers 4.5 V to 36 V

6 Specifications

6.1 Absolute Maximum Ratings

 over operating free-air temperature range (unless otherwise noted)⁽¹⁾

	MIN	MAX	UNIT
Supply voltage, V_{CC1} ⁽²⁾		36	V
Output supply voltage, V_{CC2}		36	V
Input voltage, V_I		7	V
Output voltage, V_O	-3	$V_{CC2} + 3$	V
Peak output current, I_O (nonrepetitive, $t \leq 5$ ms): L293	-2	2	A
Peak output current, I_O (nonrepetitive, $t \leq 100$ μ s): L293D	-1.2	1.2	A
Continuous output current, I_O : L293	-1	1	A
Continuous output current, I_O : L293D	-600	600	mA
Maximum junction temperature, T_J		150	°C
Storage temperature, T_{stg}	-65	150	°C

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) All voltage values are with respect to the network ground terminal.

6.2 ESD Ratings

		VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101 ⁽²⁾	
		± 2000	
		± 1000	

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
Supply voltage	V_{CC1}	4.5		7	V
	V_{CC2}	V_{CC1}		36	
V_{IH}	High-level input voltage	$V_{CC1} \leq 7$ V		V_{CC1}	V
		$V_{CC1} \geq 7$ V	2.3	7	V
V_{IL}	Low-level output voltage	-0.3 ⁽¹⁾		1.5	V
T_A	Operating free-air temperature	0		70	°C

- (1) The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

6.4 Thermal Information

THERMAL METRIC ⁽¹⁾		L293, L293D	UNIT
		NE (PDIP)	
		16 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance ⁽²⁾	36.4	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	22.5	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	16.5	°C/W
Ψ_{JT}	Junction-to-top characterization parameter	7.1	°C/W
Ψ_{JB}	Junction-to-board characterization parameter	16.3	°C/W

- (1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, [SPRA953](#).
- (2) The package thermal impedance is calculated in accordance with JESD 51-7.

6.5 Electrical Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V _{OH}	High-level output voltage	L293: I _{OH} = -1 A		V _{CC2} - 1.8	V _{CC2} - 1.4		V
		L293D: I _{OH} = -0.6 A					
V _{OL}	Low-level output voltage	L293: I _{OL} = 1 A			1.2	1.8	V
		L293D: I _{OL} = 0.6 A					
V _{OKH}	High-level output clamp voltage	L293D: I _{OK} = -0.6 A		V _{CC2} + 1.3			V
V _{OKL}	Low-level output clamp voltage	L293D: I _{OK} = 0.6 A		1.3			V
I _{IH}	High-level input current	A	V _I = 7 V	0.2		100	μA
		EN		0.2		10	
I _{IL}	Low-level input current	A	V _I = 0	-3		-10	μA
		EN		-2		-100	
I _{CC1}	Logic supply current	I _O = 0	All outputs at high level	13		22	mA
			All outputs at low level	35		60	
			All outputs at high impedance	8		24	
I _{CC2}	Output supply current	I _O = 0	All outputs at high level	14		24	mA
			All outputs at low level	2		6	
			All outputs at high impedance	2		4	

6.6 Switching Characteristics

over operating free-air temperature range (unless otherwise noted) V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t _{PLH}	Propagation delay time, low-to-high-level output from A input	L293NE, L293DNE		800			ns
		L293DWP, L293N L293DN		750			
t _{PHL}	Propagation delay time, high-to-low-level output from A input	L293NE, L293DNE		400			ns
		L293DWP, L293N L293DN		200			
t _{TLH}	Transition time, low-to-high-level output	L293NE, L293DNE		300			ns
		L293DWP, L293N L293DN		100			
t _{THL}	Transition time, high-to-low-level output	L293NE, L293DNE		300			ns
		L293DWP, L293N L293DN		350			

6.7 Typical Characteristics

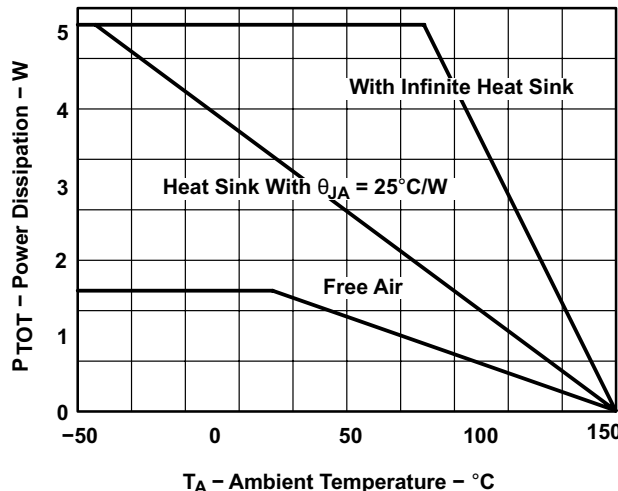
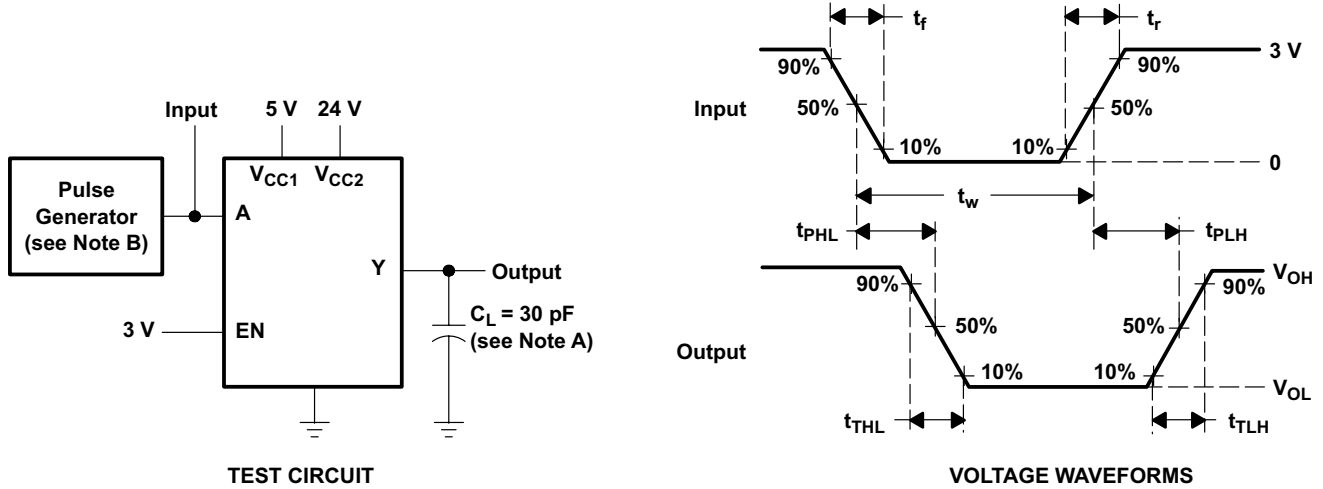


Figure 1. Maximum Power Dissipation vs Ambient Temperature

7 Parameter Measurement Information



NOTES: A. C_L includes probe and jig capacitance.

B. The pulse generator has the following characteristics: $t_r \leq 10$ ns, $t_f \leq 10$ ns, $t_w = 10$ μ s, PRR = 5 kHz, $Z_O = 50$ Ω .

Figure 2. Test Circuit and Voltage Waveforms

8 Detailed Description

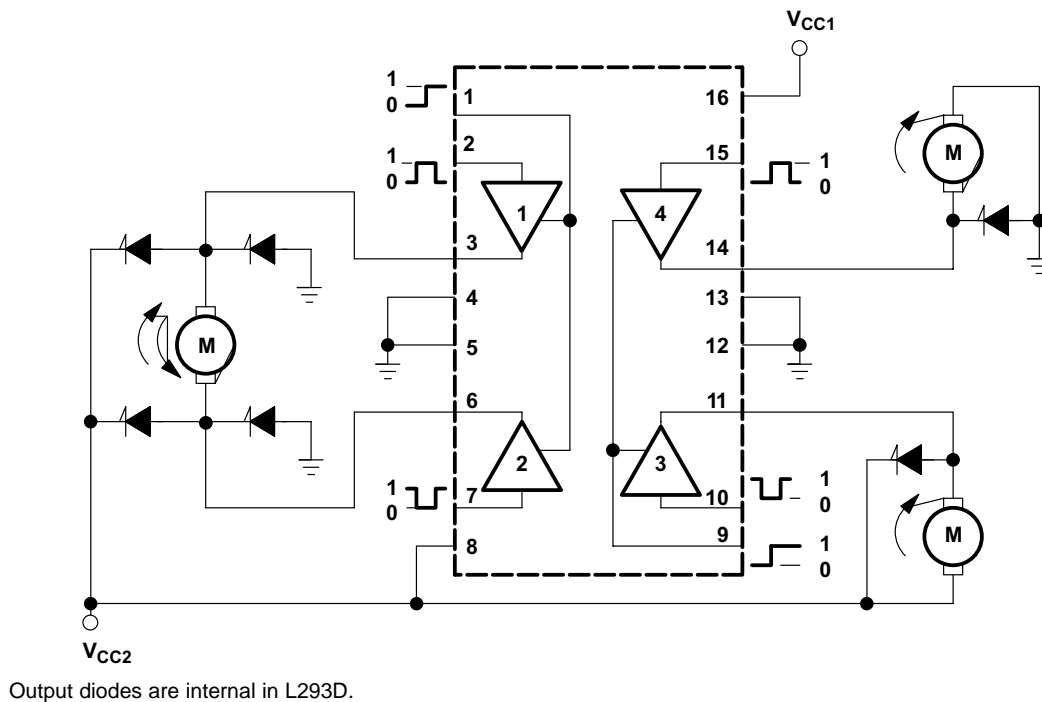
8.1 Overview

The L293 and L293D are quadruple high-current half-H drivers. These devices are designed to drive a wide array of inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current and high-voltage loads. All inputs are TTL compatible and tolerant up to 7 V.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. On the L293D, these diodes are integrated to reduce system complexity and overall system size. A V_{CC1} terminal, separate from V_{CC2} , is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0°C to 70°C.

8.2 Functional Block Diagram



8.3 Feature Description

The L293x has TTL-compatible inputs and high voltage outputs for inductive load driving. Current outputs can get up to 2 A using the L293.

8.4 Device Functional Modes

Table 1 lists the functional modes of the L293x.

Table 1. Function Table (Each Driver)⁽¹⁾

INPUTS ⁽²⁾		OUTPUT (Y)
A	EN	
H	H	H
L	H	L
X	L	Z

- (1) H = high level, L = low level, X = irrelevant, Z = high impedance (off)
- (2) In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.

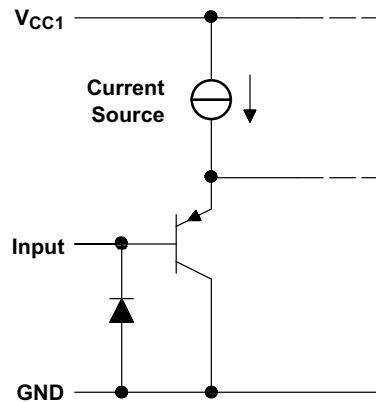


Figure 3. Schematic of Inputs for the L293x

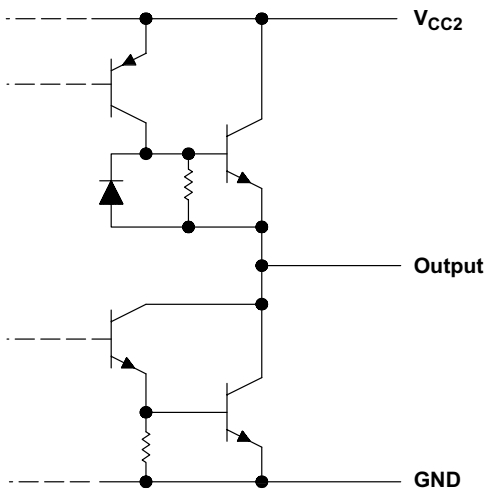


Figure 4. Schematic of Outputs for the L293

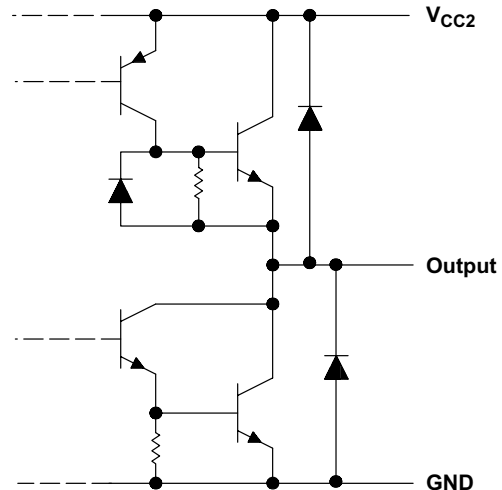


Figure 5. Schematic of Outputs for the L293D

9 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

9.1 Application Information

A typical application for the L293 device is driving a two-phase motor. Below is an example schematic displaying how to properly connect a two-phase motor to the L293 device.

Provide a 5-V supply to V_{CC1} and valid logic input levels to data and enable inputs. V_{CC2} must be connected to a power supply capable of supplying the needed current and voltage demand for the loads connected to the outputs.

9.2 Typical Application

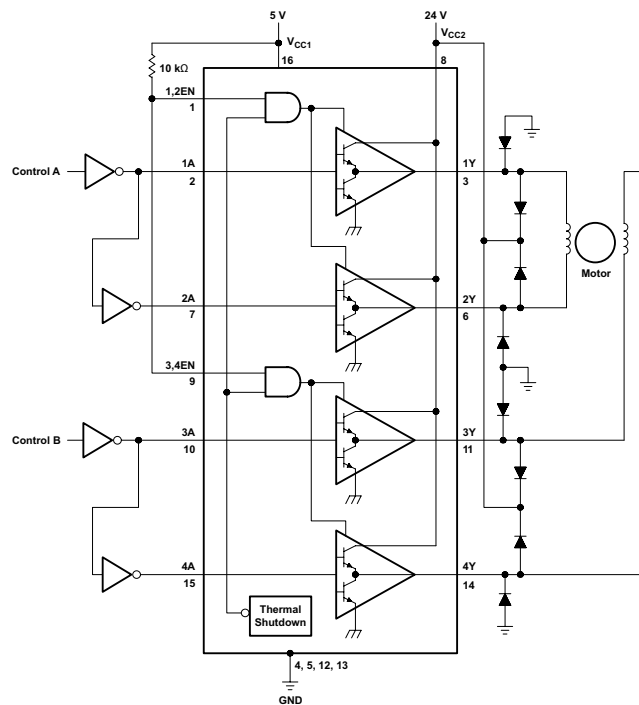


Figure 6. Two-Phase Motor Driver (L293)

9.2.1 Design Requirements

The design techniques in the application above as well as the applications below should fall within the following design requirements.

1. V_{CC1} should fall within the limits described in the [Recommended Operating Conditions](#).
2. V_{CC2} should fall within the limits described in the [Recommended Operating Conditions](#).
3. The current per channel should not exceed 1 A for the L293 (600mA for the L293D).

9.2.2 Detailed Design Procedure

When designing with the L293 or L293D, careful consideration should be made to ensure the device does not exceed the operating temperature of the device. Proper heatsinking will allow for operation over a larger range of current per channel. Refer to the [Power Supply Recommendations](#) as well as the [Layout Example](#).

Typical Application (continued)

9.2.3 Application Curve

Refer to [Power Supply Recommendations](#) for additional information with regards to appropriate power dissipation. [Figure 7](#) describes thermal dissipation based on [Figure 14](#).

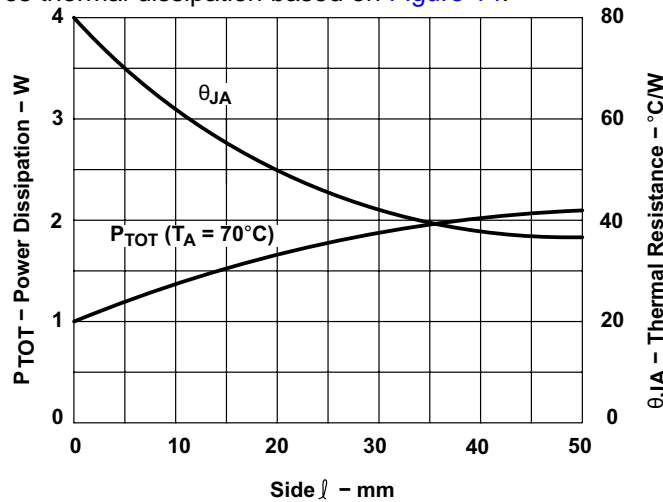


Figure 7. Maximum Power and Junction vs Thermal Resistance

9.3 System Examples

9.3.1 L293D as a Two-Phase Motor Driver

[Figure 8](#) below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the [Recommended Operating Conditions](#) when considering the appropriate input high and input low voltage levels to enable each channel of the device.

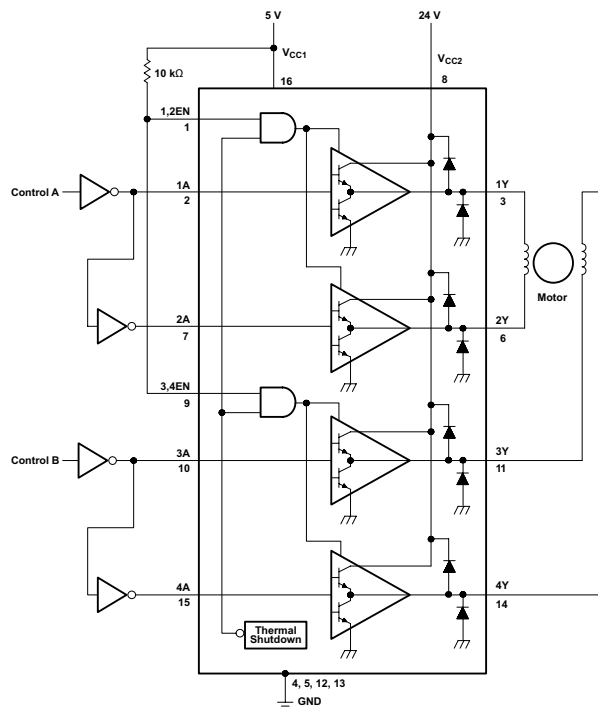
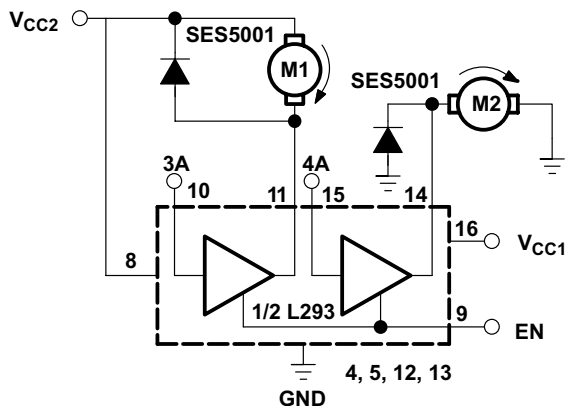


Figure 8. Two-Phase Motor Driver (L293D)

System Examples (continued)

9.3.2 DC Motor Controls

Figure 9 and Figure 10 below depict a typical setup for using the L293 device as a controller for DC motors. Note that the L293 device can be used as a simple driver for a motor to turn on and off in one direction, and can also be used to drive a motor in both directions. Refer to the function tables below to understand unidirectional vs bidirectional motor control. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.



Connections to ground and to supply voltage

Figure 9. DC Motor Controls

Table 2. Unidirectional DC Motor Control

EN	3A	M1 ⁽¹⁾	4A	M2
H	H	Fast motor stop	H	Run
H	L	run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

(1) L = low, H = high, X = don't care

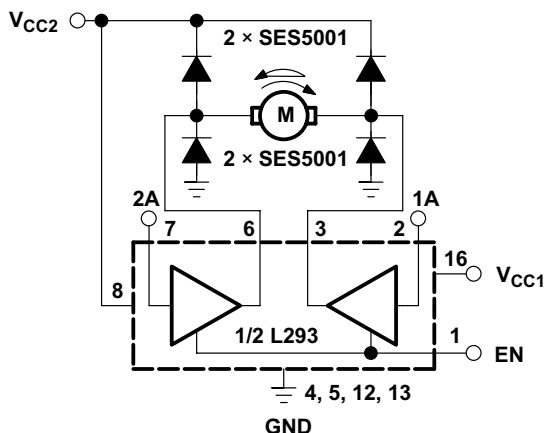


Figure 10. Bidirectional DC Motor Control

Table 3. Bidirectional DC Motor Control

EN	1A	2A	FUNCTION ⁽¹⁾
H	L	H	Turn right
H	H	L	Turn left

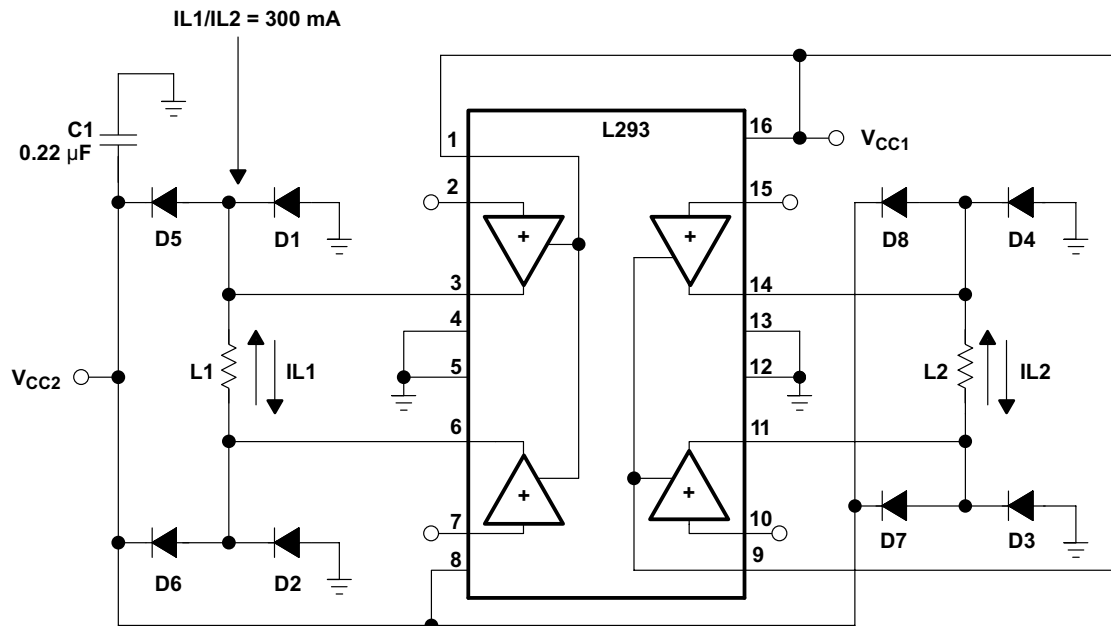
(1) L = low, H = high, X = don't care

Table 3. Bidirectional DC Motor Control (continued)

EN	1A	2A	FUNCTION ⁽¹⁾
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Free-running motor stop

9.3.3 Bipolar Stepping-Motor Control

Figure 11 below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the [Recommended Operating Conditions](#) when considering the appropriate input high and input low voltage levels to enable each channel of the device.



D1–D8 = SES5001

Figure 11. Bipolar Stepping-Motor Control

10 Power Supply Recommendations

V_{CC1} is $5\text{ V} \pm 0.5\text{ V}$ and V_{CC2} can be same supply as V_{CC1} or a higher voltage supply with peak voltage up to 36 V. Bypass capacitors of 0.1 μF or greater should be used at V_{CC1} and V_{CC2} pins. There are no power up or power down supply sequence order requirements.

Properly heatsinking the L293 when driving high-current is critical to design. The $R_{thj-amp}$ of the L293 can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board or to an external heat sink.

Figure 14 shows the maximum package power $PTOT$ and the θ_{JA} as a function of the side of two equal square copper areas having a thickness of 35 μm (see Figure 14). In addition, an external heat sink can be used (see Figure 12).

During soldering, the pin temperature must not exceed 260°C, and the soldering time must not exceed 12 seconds.

The external heatsink or printed circuit copper area must be connected to electrical ground.

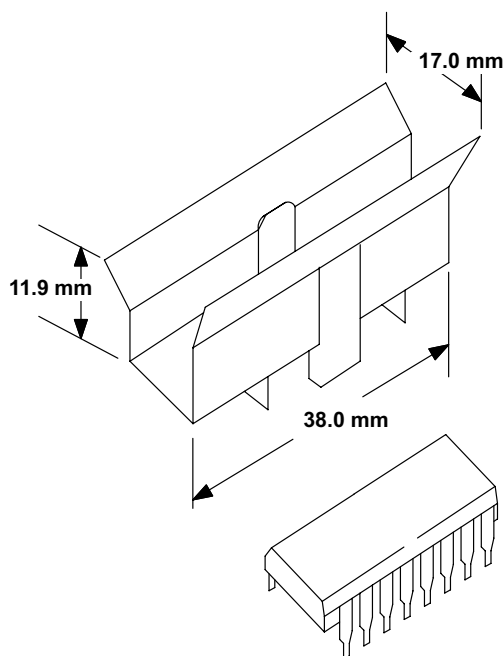


Figure 12. External Heat Sink Mounting Example ($\theta_{JA} = 25^\circ\text{C/W}$)

11 Layout

11.1 Layout Guidelines

Place the device near the load to keep output traces short to reduce EMI. Use solid vias to transfer heat from ground pins to ground plane of the printed-circuit-board.

11.2 Layout Example

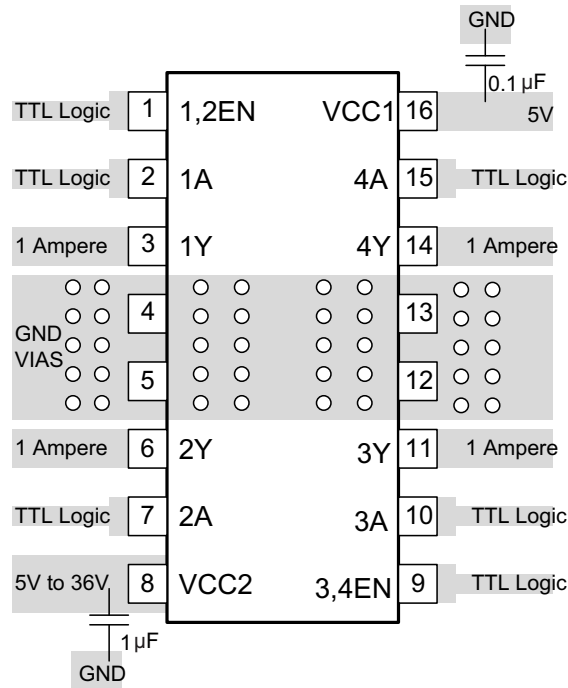


Figure 13. Layout Diagram

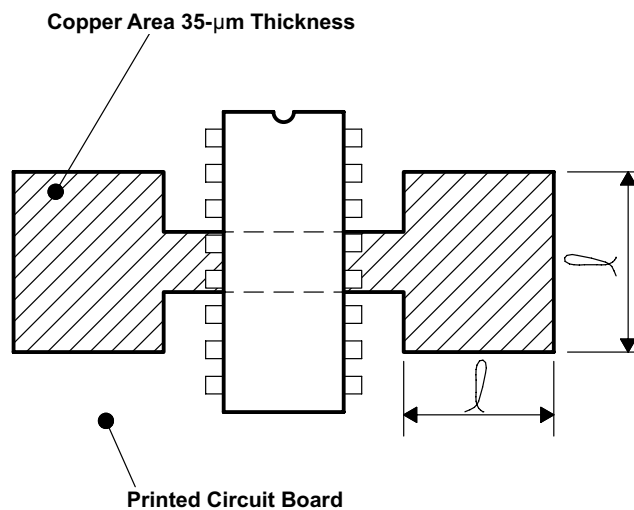


Figure 14. Example of Printed-Circuit-Board Copper Area (Used as Heat Sink)

12 Device and Documentation Support

12.1 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to sample or buy.

Table 4. Related Links

PARTS	PRODUCT FOLDER	SAMPLE & BUY	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
L293	Click here	Click here	Click here	Click here	Click here
L293D	Click here	Click here	Click here	Click here	Click here

12.2 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's [Terms of Use](#).

TI E2E™ Online Community *TI's Engineer-to-Engineer (E2E) Community*. Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

Design Support *TI's Design Support* Quickly find helpful E2E forums along with design support tools and contact information for technical support.

12.3 Trademarks

E2E is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

12.4 Electrostatic Discharge Caution



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

12.5 Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
L293DNE	ACTIVE	PDIP	NE	16	25	RoHS & Green	NIPDAU	N / A for Pkg Type	0 to 70	L293DNE	Samples
L293DNEE4	ACTIVE	PDIP	NE	16	25	RoHS & Green	NIPDAU	N / A for Pkg Type	0 to 70	L293DNE	Samples
L293NE	ACTIVE	PDIP	NE	16	25	RoHS & Green	NIPDAU	N / A for Pkg Type	0 to 70	L293NE	Samples
L293NEE4	ACTIVE	PDIP	NE	16	25	RoHS & Green	NIPDAU	N / A for Pkg Type	0 to 70	L293NE	Samples

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".

RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.

Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "-" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and

continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (<https://www.ti.com/legal/termsofsale.html>) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2021, Texas Instruments Incorporated



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Documento nº2:
Planos

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Esteso

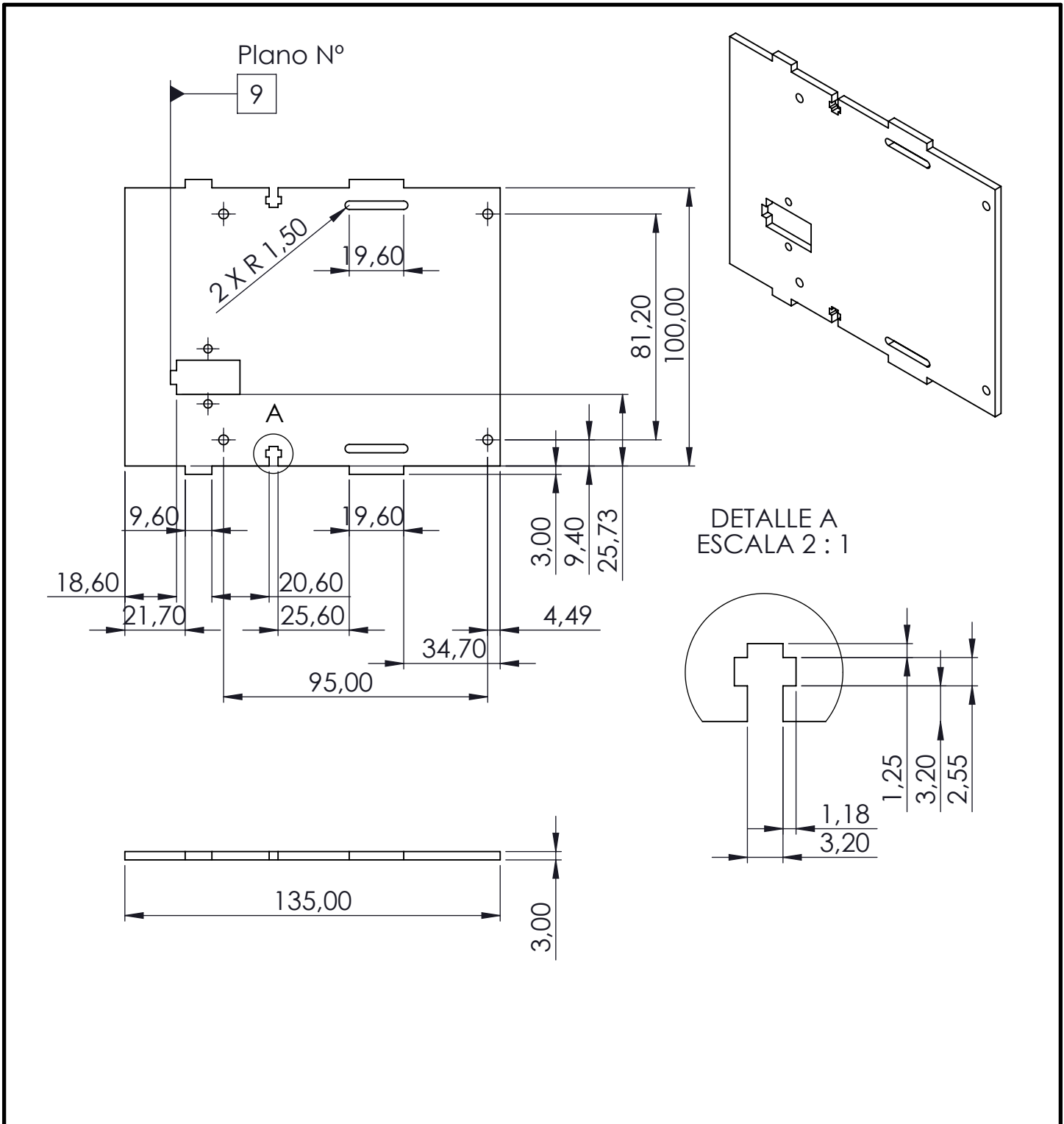
TUTORIZADO POR

Leopoldo Armesto Ángel

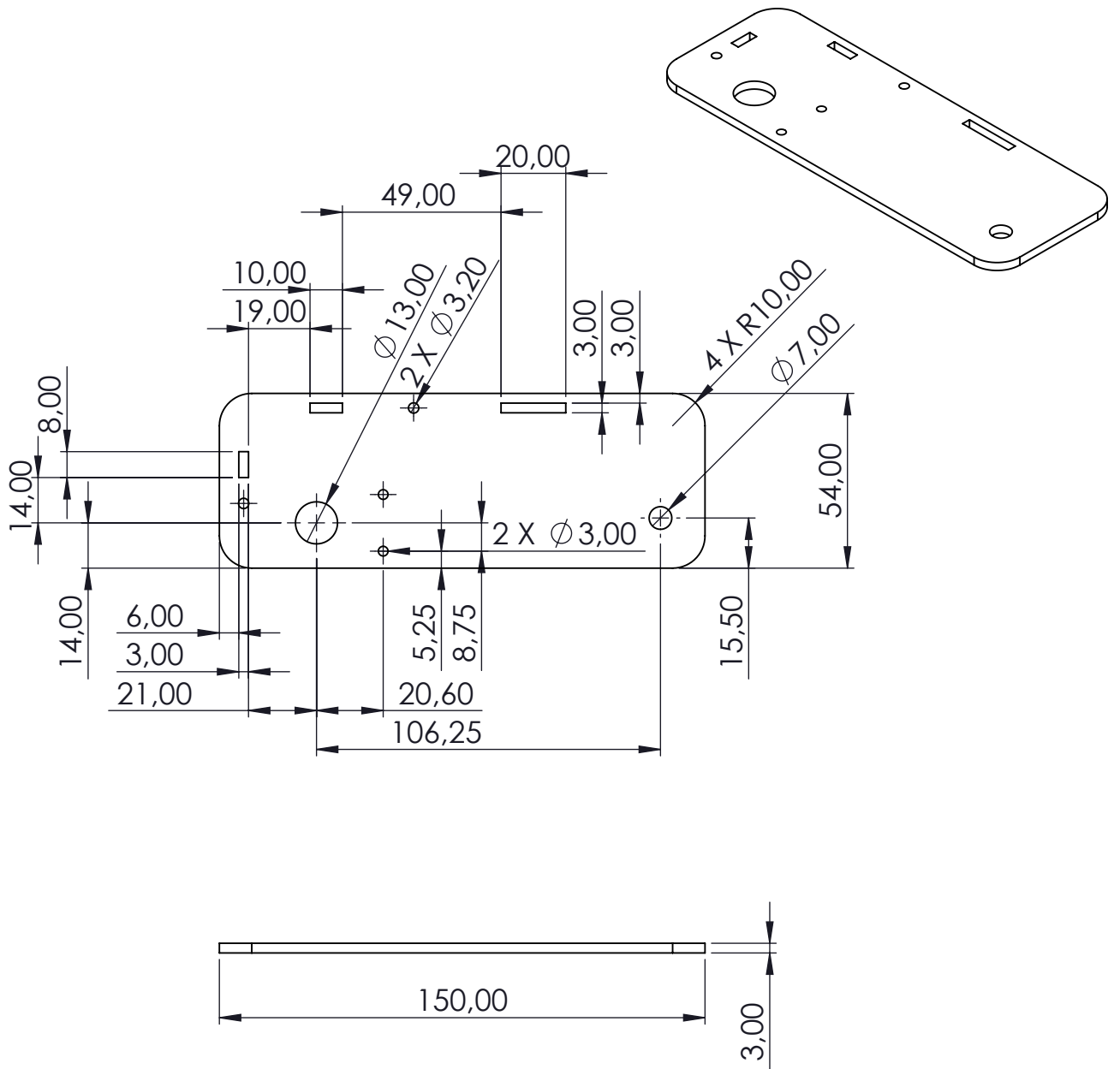
CURSO ACADÉMICO: 2020/2021




Índice de los planos

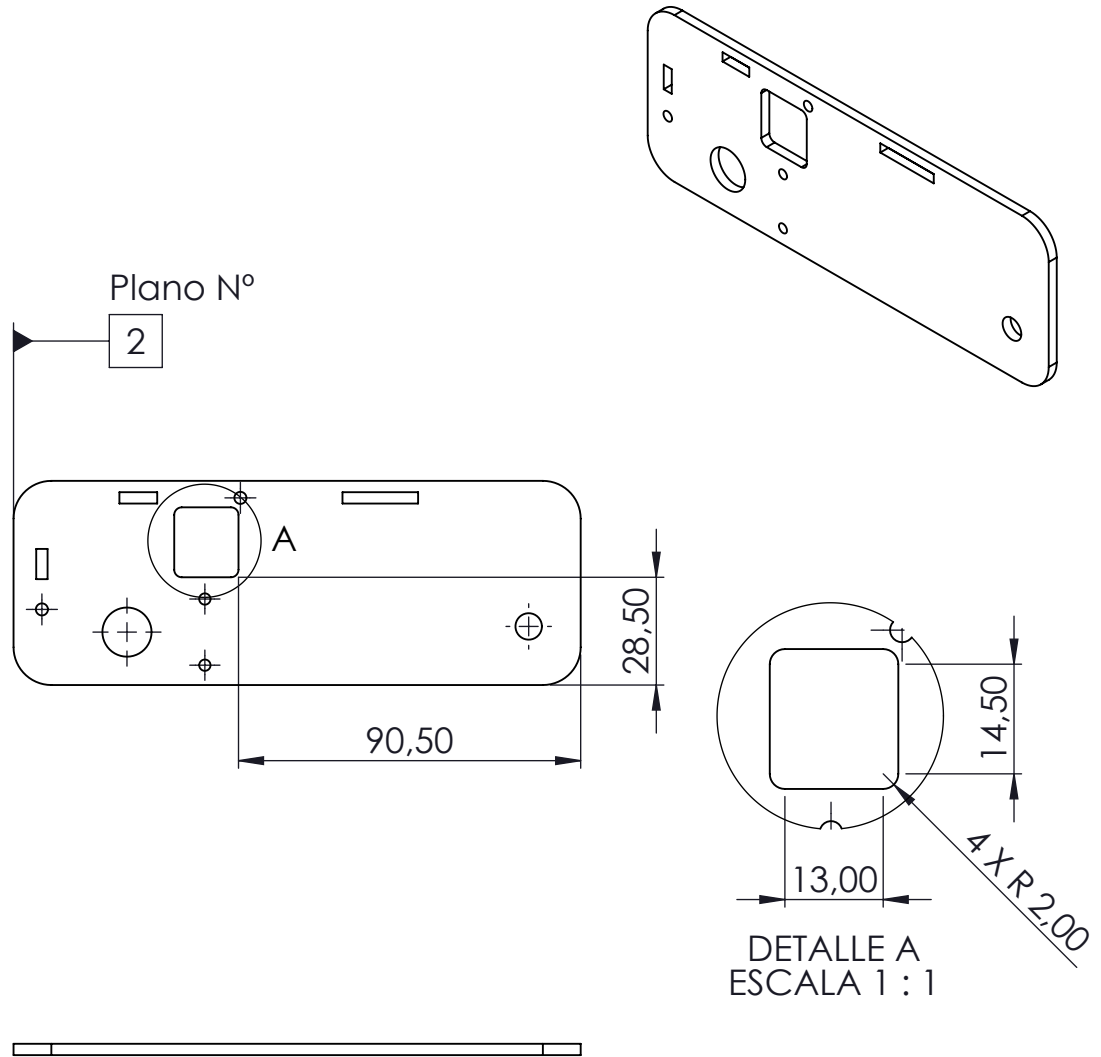
1. Base de la parte móvil.
2. Lateral derecho de la parte móvil.
3. Lateral izquierdo de la parte móvil.
4. Pieza frontal de la parte móvil.
5. Pieza nº1 de la parte de manipulador robótico.
6. Pieza nº2 de la parte de manipulador robótico.
7. Pieza nº3 de la parte de manipulador robótico.
8. Pieza nº4 de la parte de manipulador robótico.
9. Pieza nº5 de la parte de manipulador robótico.
10. Pieza nº6 de la parte de manipulador robótico.
11. Pieza nº7 de la parte de manipulador robótico.
12. Pieza nº8 de la parte de manipulador robótico.
13. Pieza nº9 de la parte de manipulador robótico.
14. Pieza nº10 de la parte de manipulador robótico.
15. Pieza nº11 de la parte de manipulador robótico.
16. Pieza nº12 de la parte de manipulador robótico.
17. Pieza nº13 de la parte de manipulador robótico.
18. Pieza nº14 de la parte de manipulador robótico.
19. Pieza nº15 de la parte de manipulador robótico.
20. Pieza nº16 de la parte de manipulador robótico.
21. Pieza nº17 de la parte de manipulador robótico.
22. Pieza nº18 de la parte de manipulador robótico.
23. Pieza nº19 de la parte de manipulador robótico.
24. Pieza nº20 de la parte de manipulador robótico.
25. Pieza nº21 de la parte de manipulador robótico.
26. Pieza nº22 de la parte de manipulador robótico.
27. Pieza nº23 de la parte de manipulador robótico.
28. Pieza nº24 de la parte de manipulador robótico.
29. Pieza nº25 de la parte de manipulador robótico.
30. Pieza nº26 de la parte de manipulador robótico.
31. Pieza nº27 de la parte de manipulador robótico.
32. Ensamblaje de la parte móvil.
33. Ensamblaje de la pala del manipulador.
34. Ensamblaje de la cabina del manipulador.

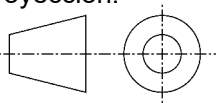




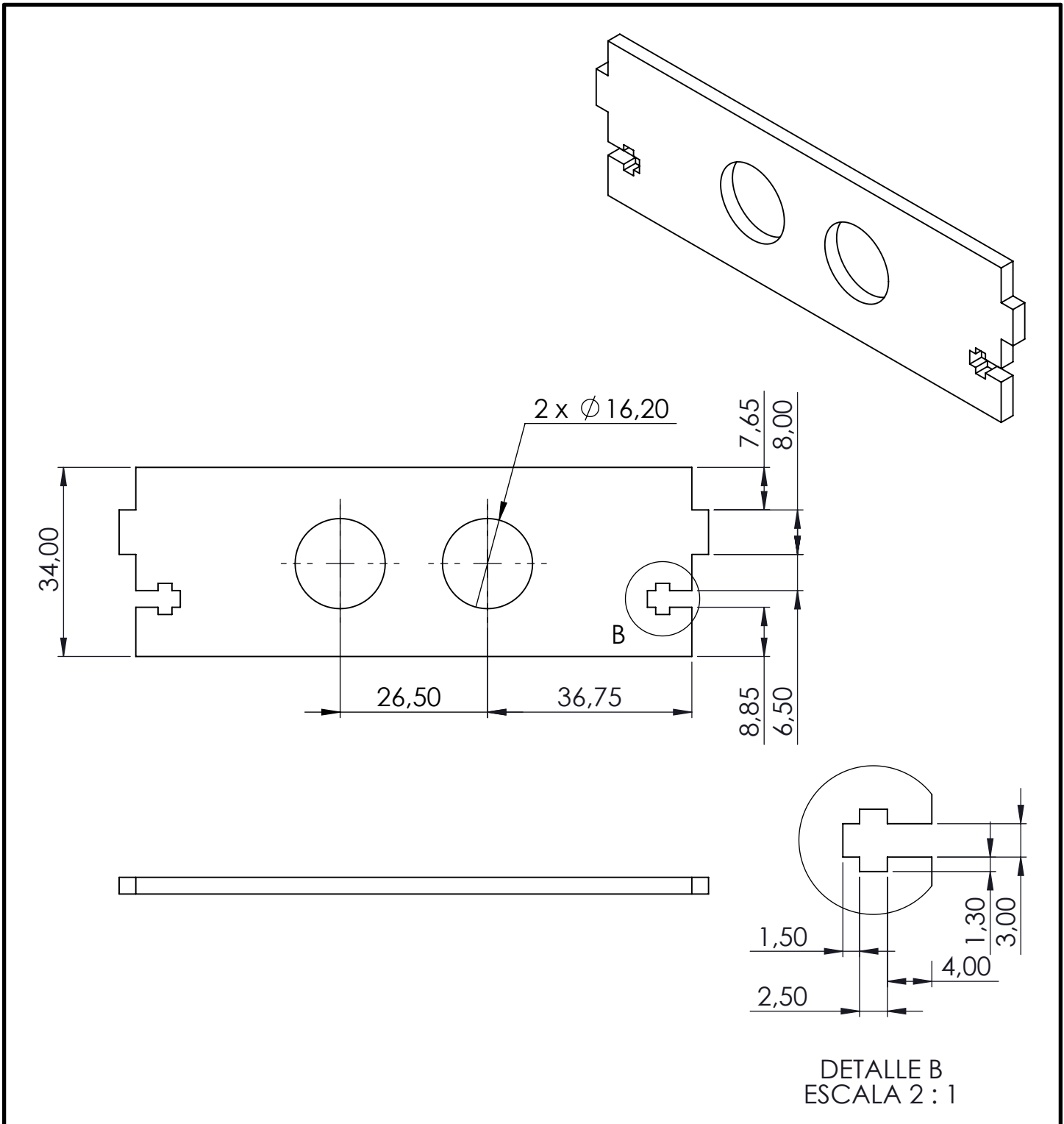
Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela: 	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 1:2
Universidad: 	Autor: David Martínez Esteso Formato: A4	TÍTULO: Base de la parte móvil	Plano N° 01

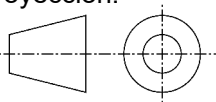




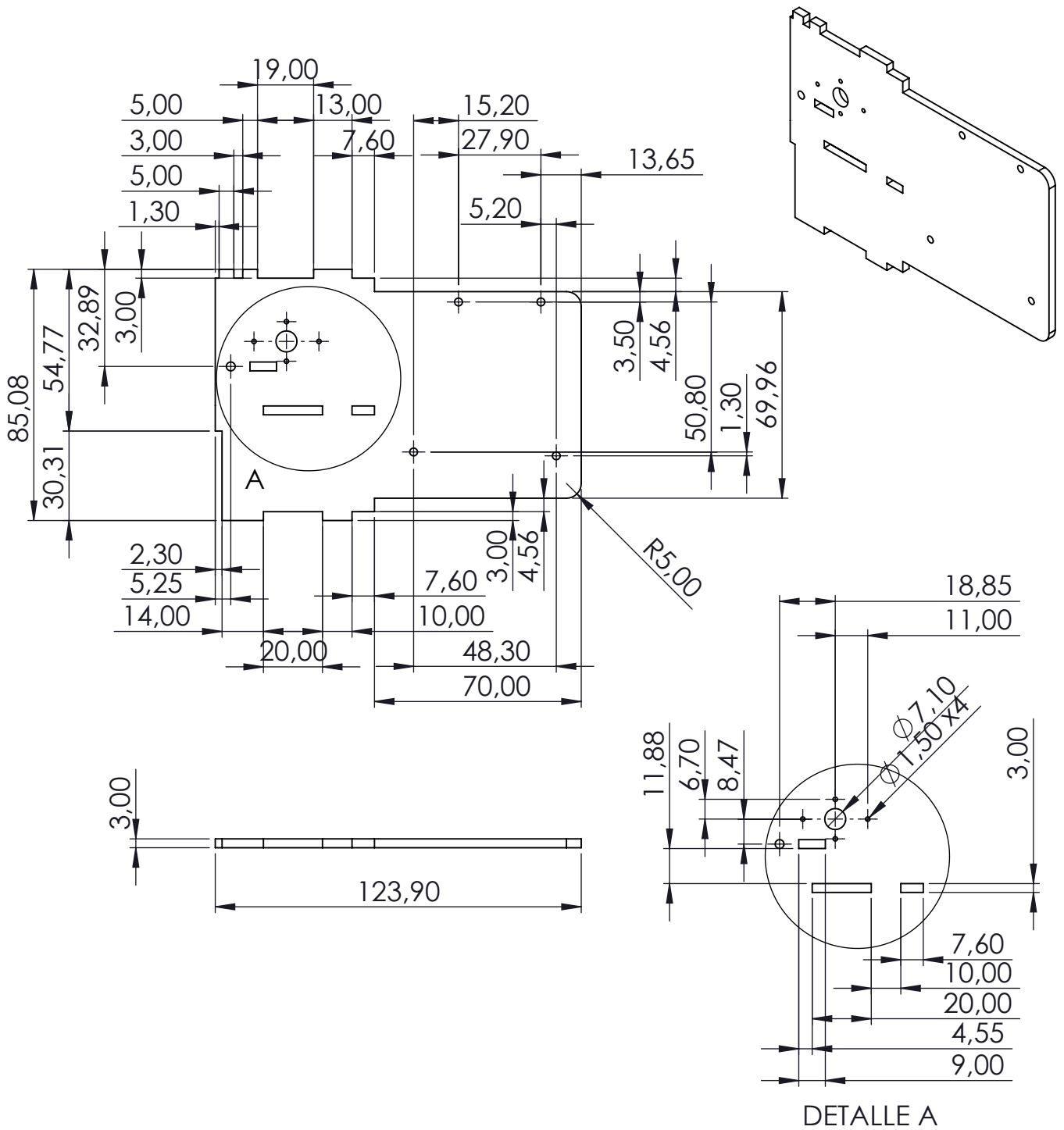
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>	<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>	<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:2</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Lateral derecho de la parte móvil</p>
<p>Plano N°</p> <p>02</p>		

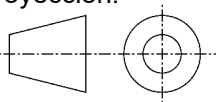




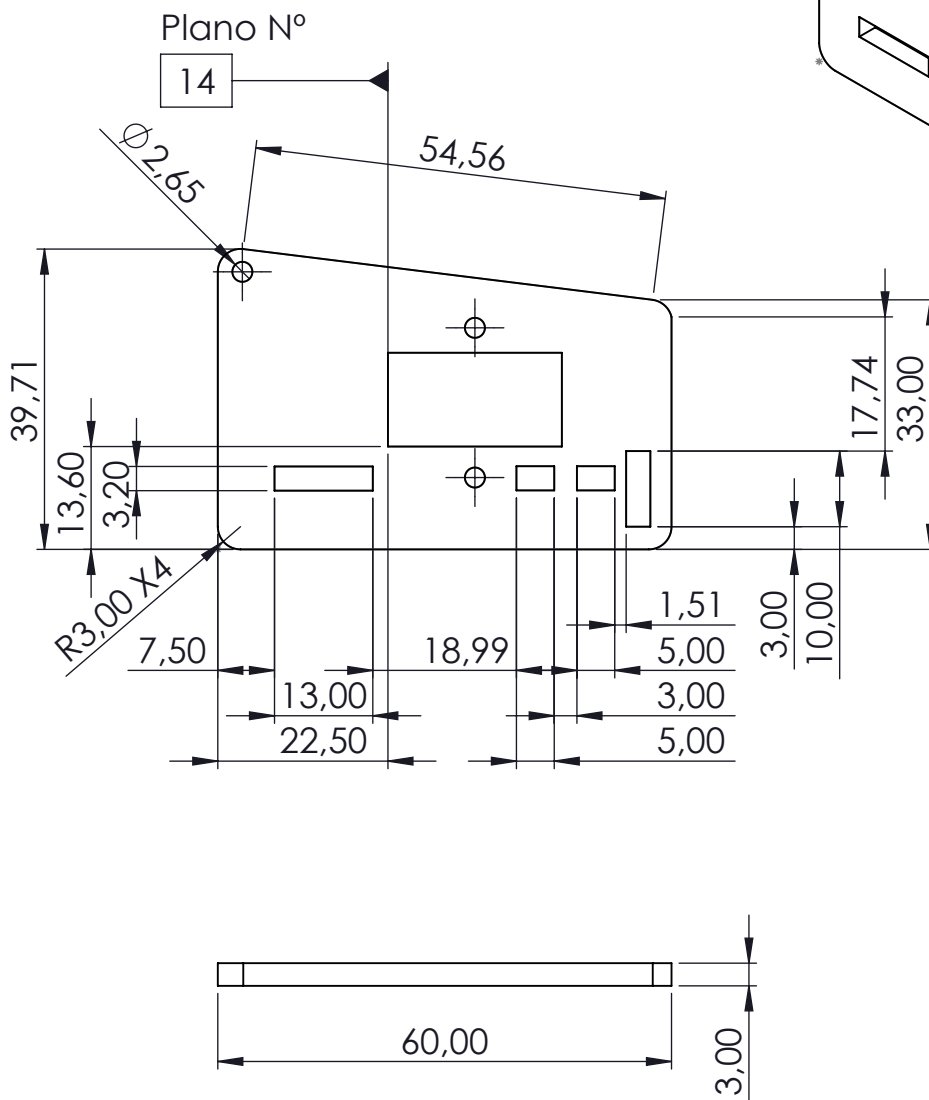
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:2</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Lateral izquierdo de la parte móvil</p>	<p>Plano N°</p> <p>03</p>

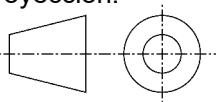




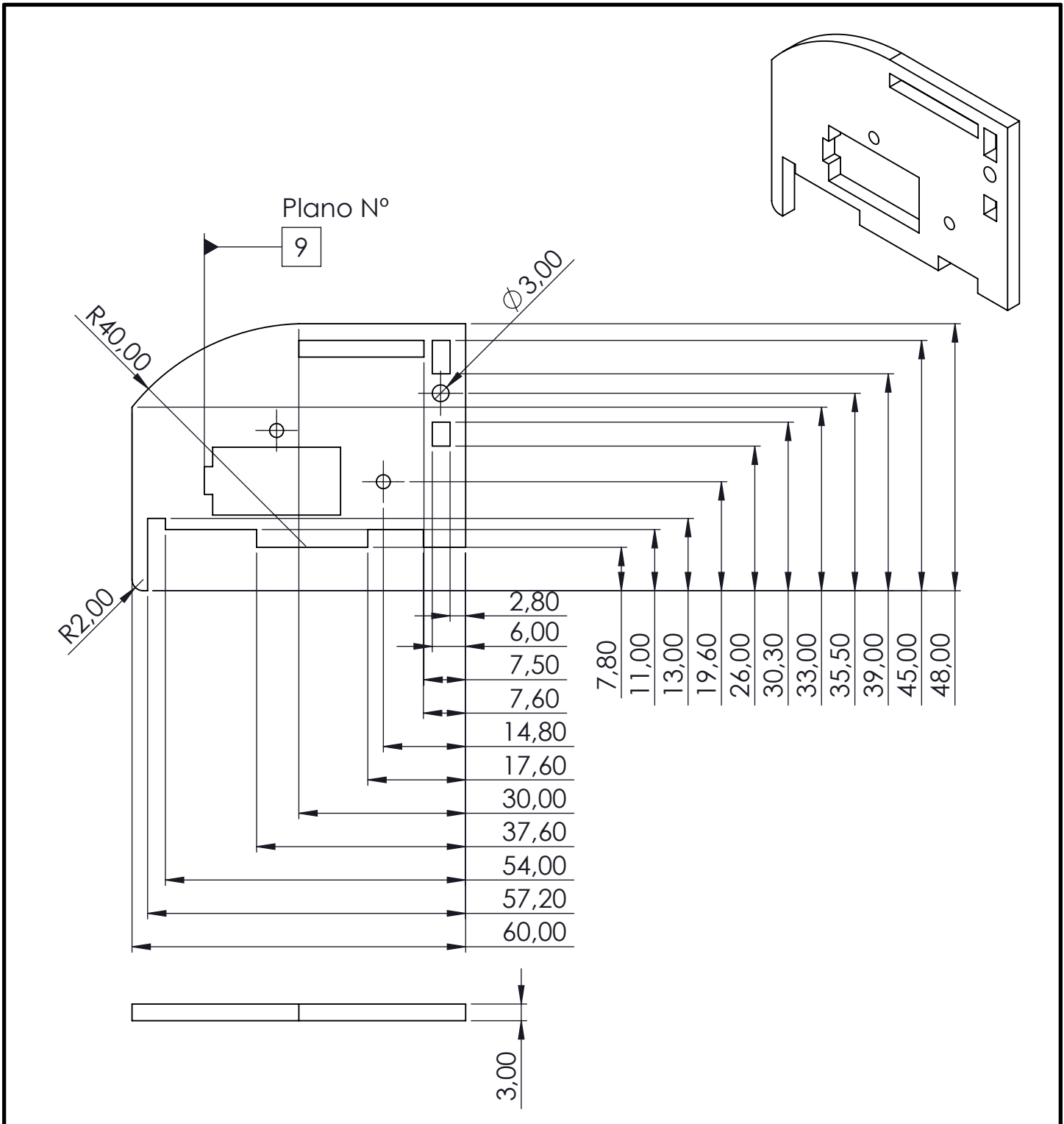
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>	<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>	<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza frontal de la parte móvil</p>
<p>Plano N°</p> <p>04</p>		

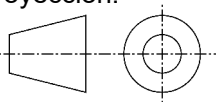




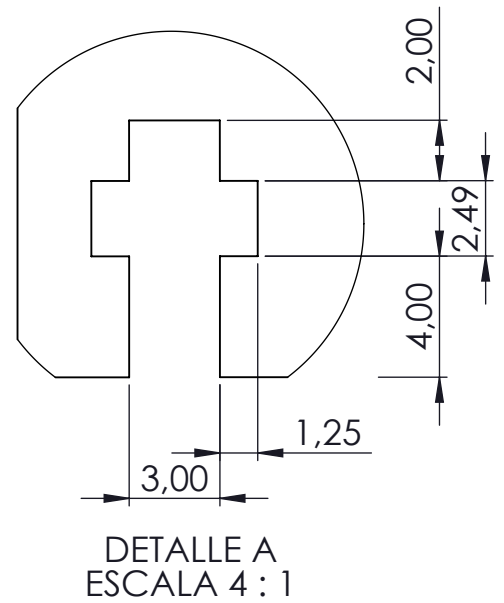
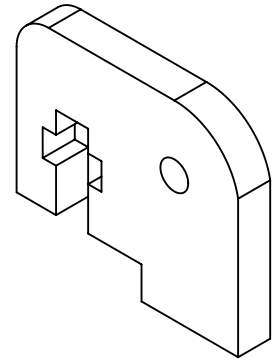
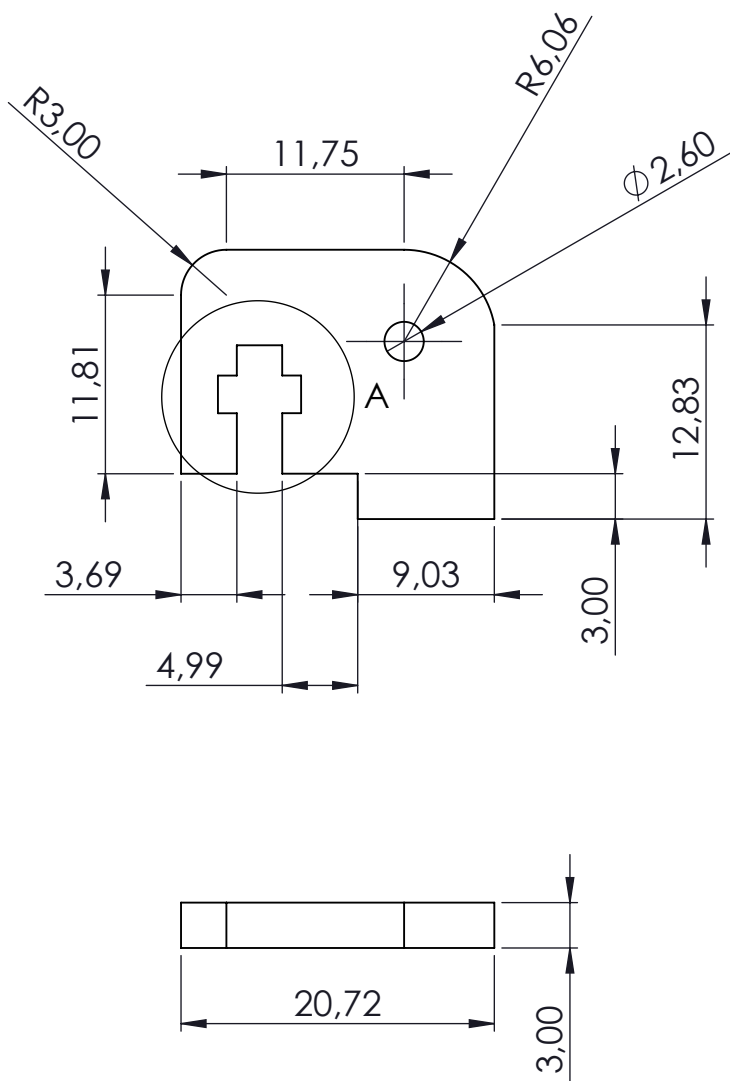
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:2</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza nº1 de la parte de manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>05</p>

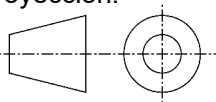




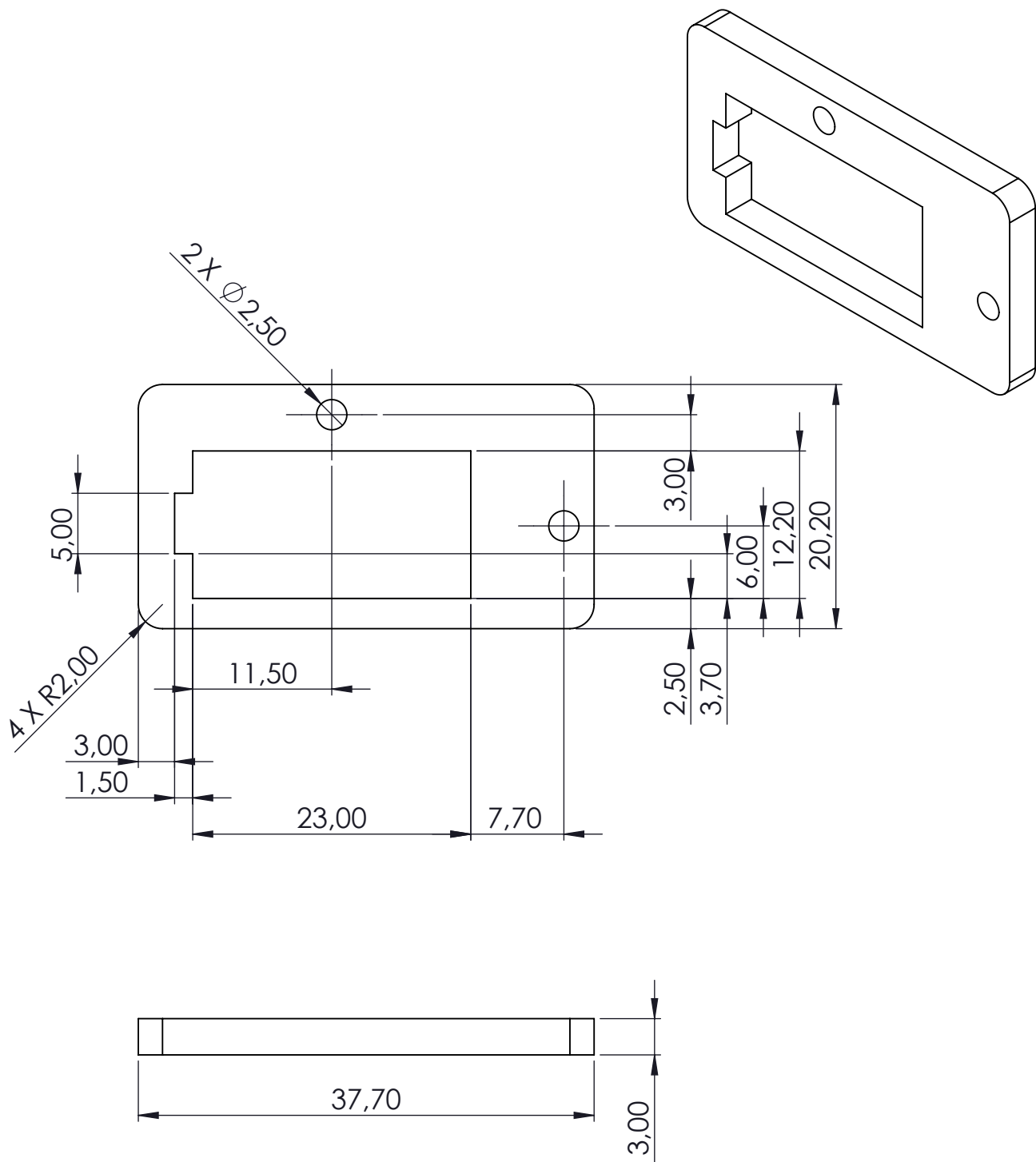
Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 1:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza nº2 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano N° 06

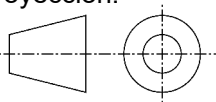




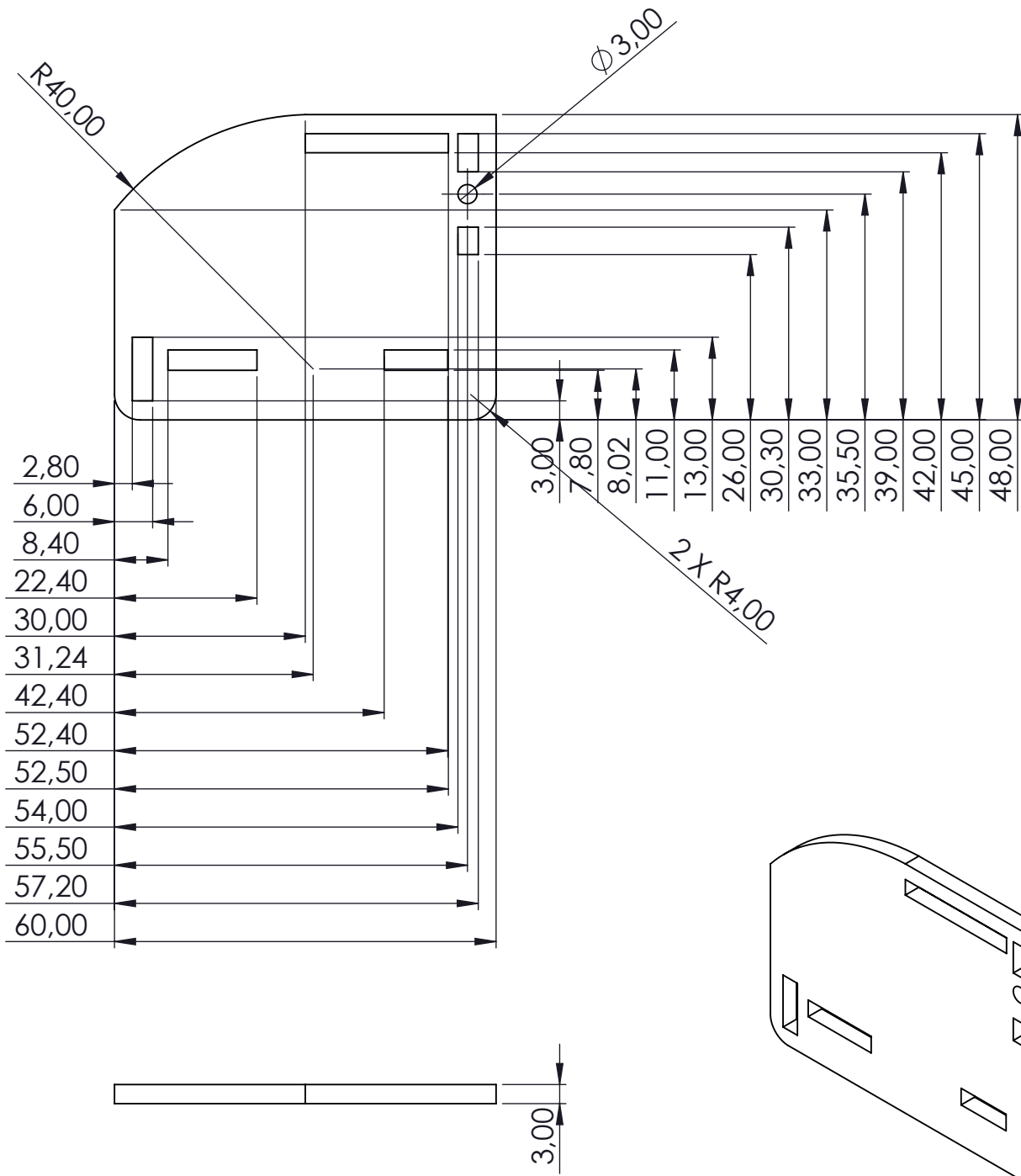
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza nº3 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano N°</p> <p>07</p>

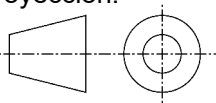




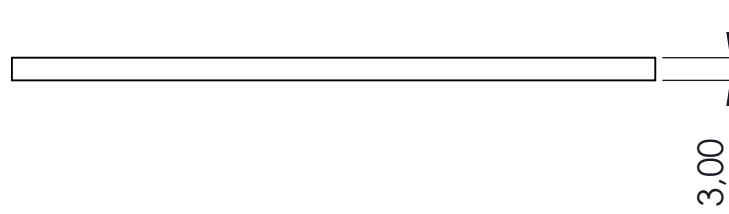
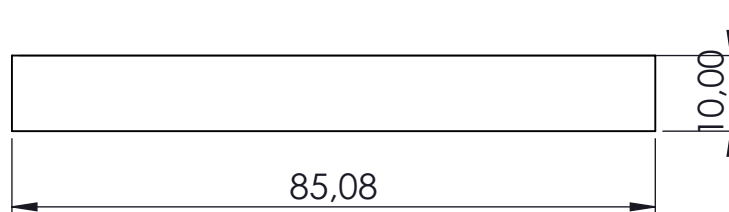
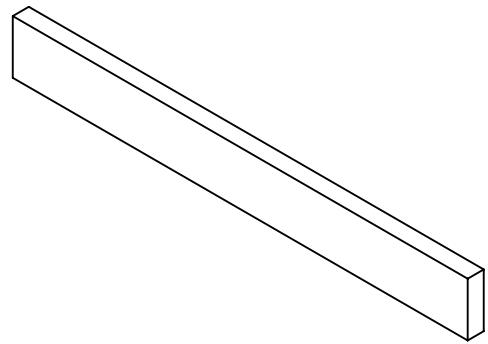
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza nº4 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>08</p>



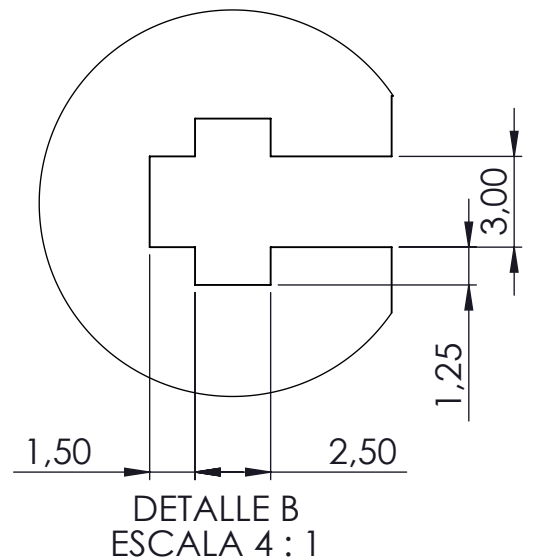
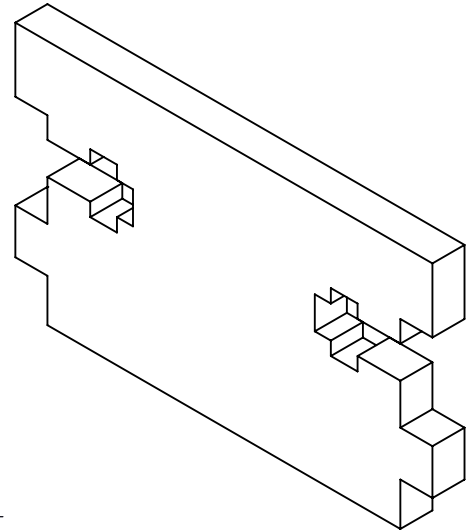
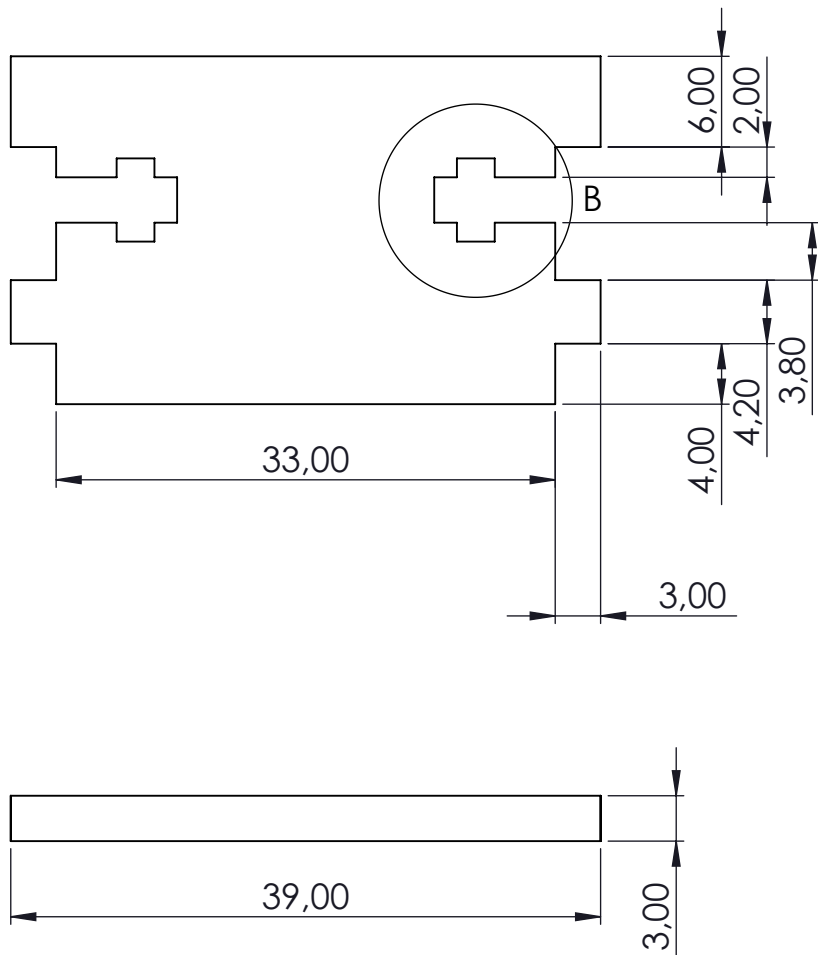
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza nº5 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>09</p>



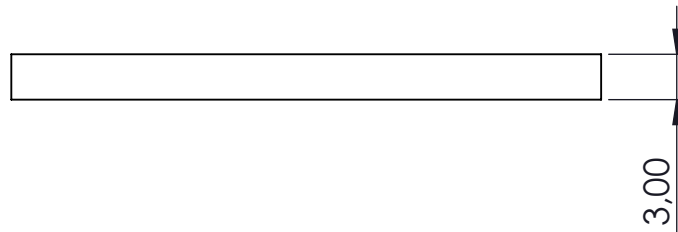
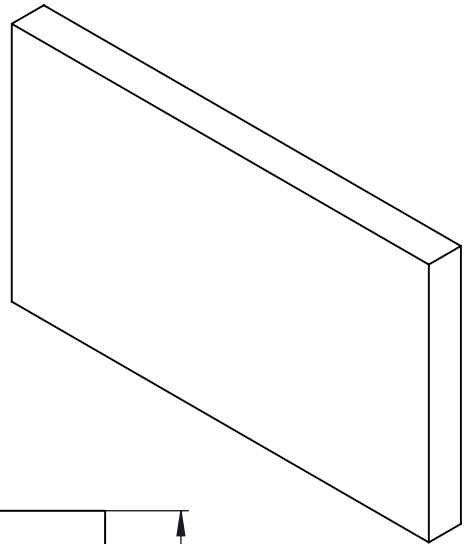
Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora	Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1	Fecha: 25/06/21 Escala: 1:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza nº6 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel
Plano N° 10		



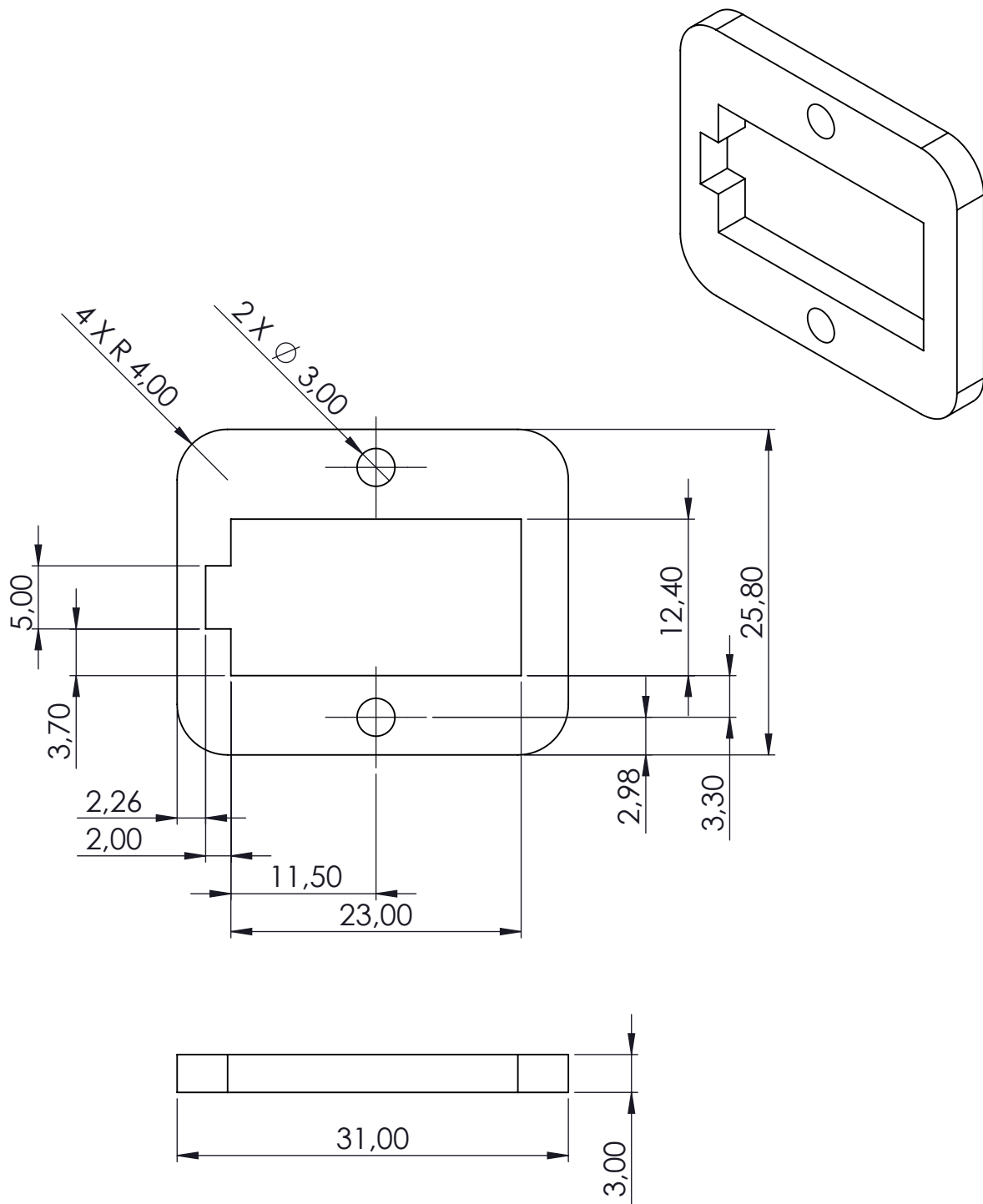
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza nº7 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>11</p>

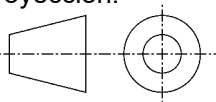




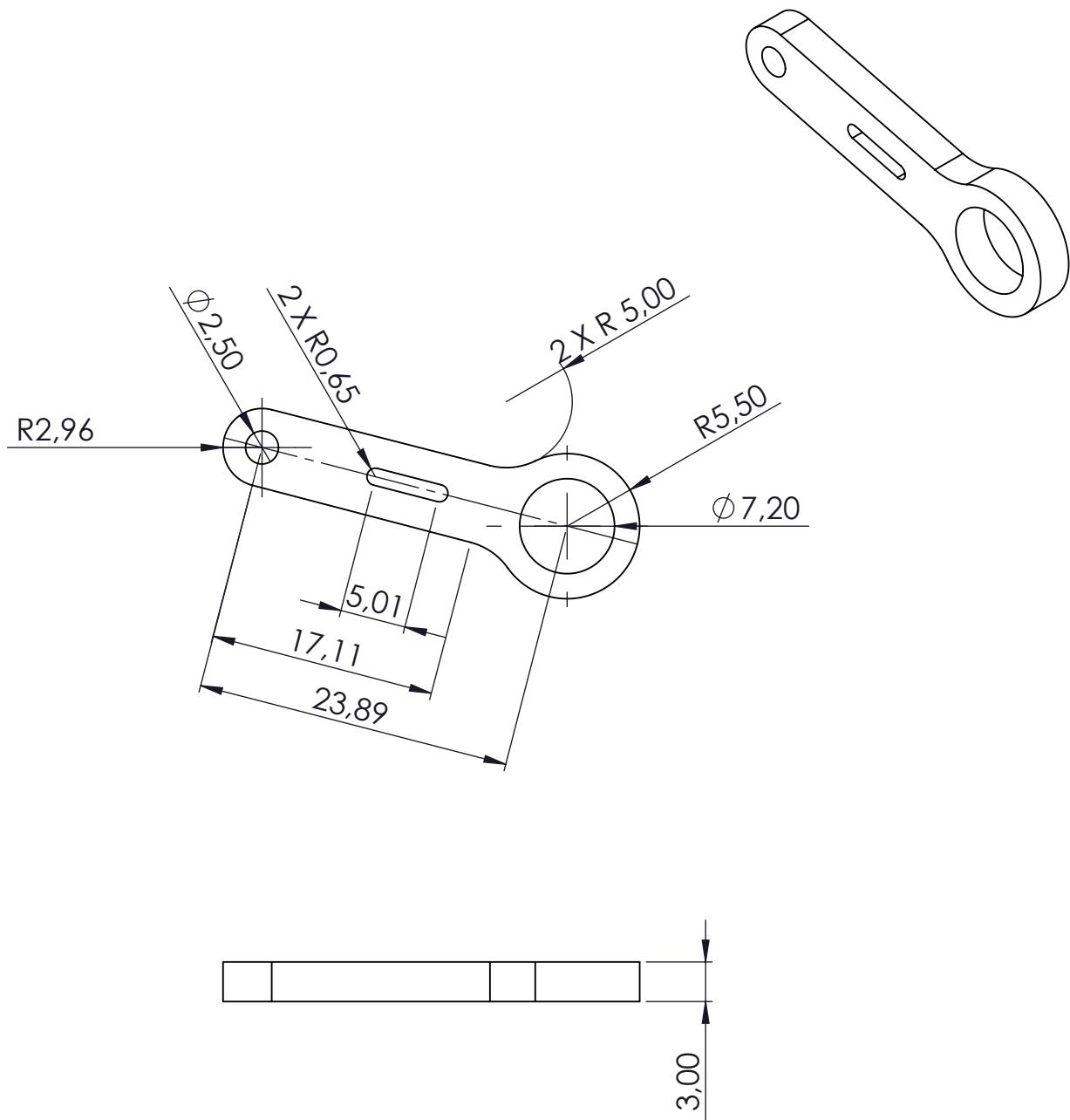
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza nº8 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>12</p>

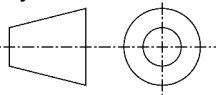




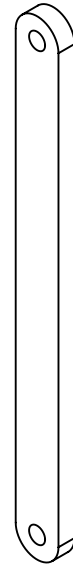
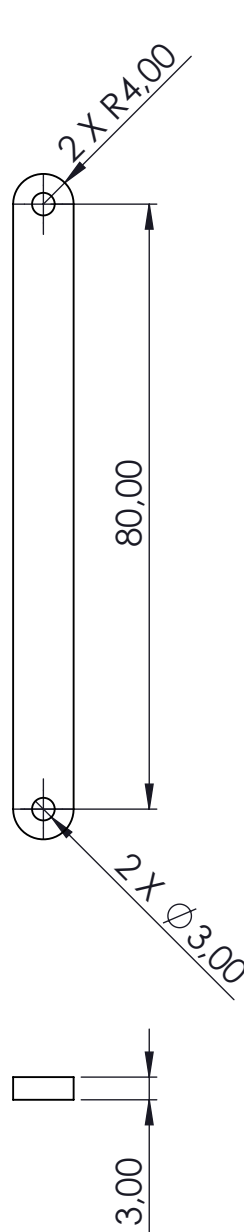
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza nº9 del brazo manipulador robótico. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>13</p>

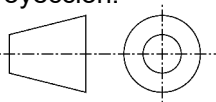




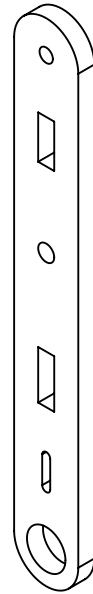
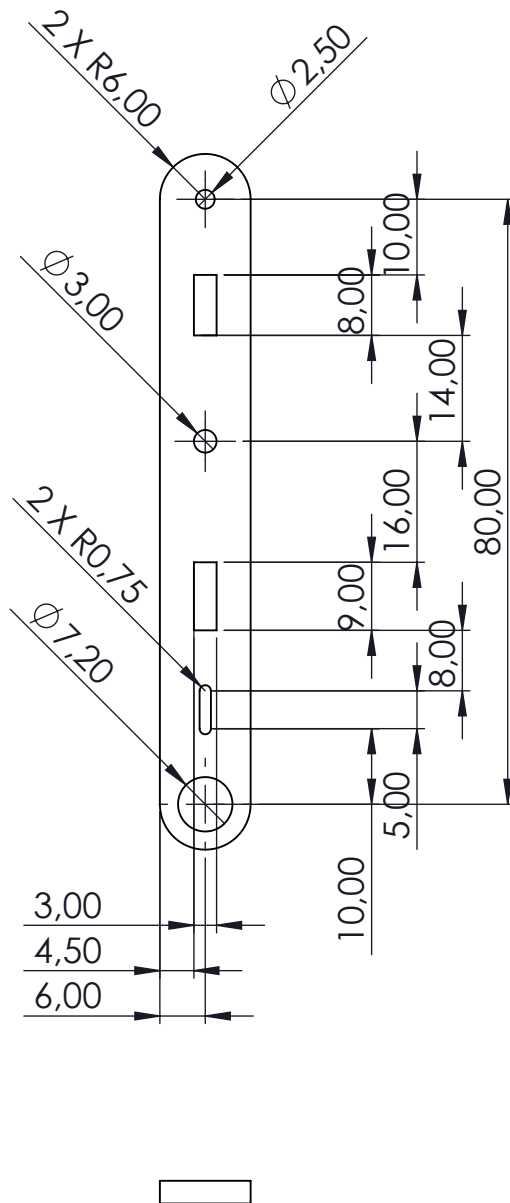
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº10. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>14</p>



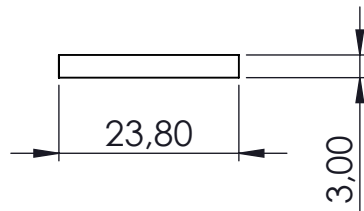
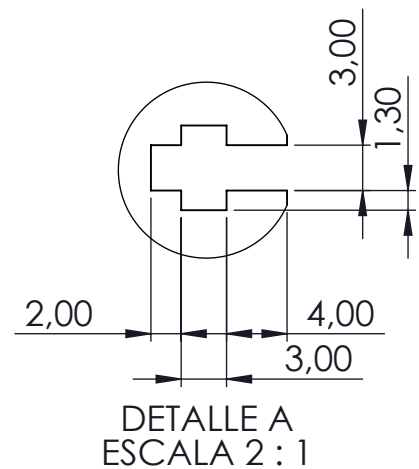
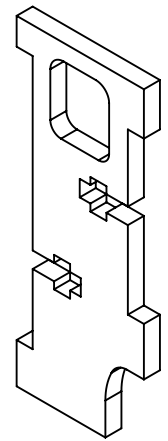
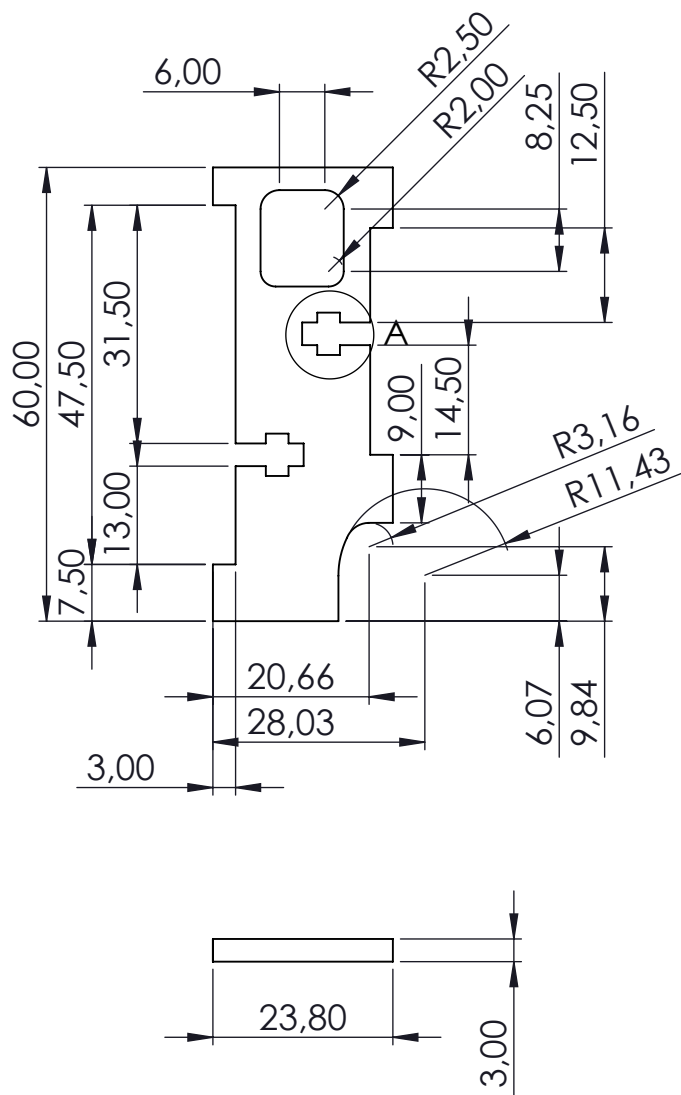
Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 2:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza del brazo manipulador robótico nº11. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano Nº 15



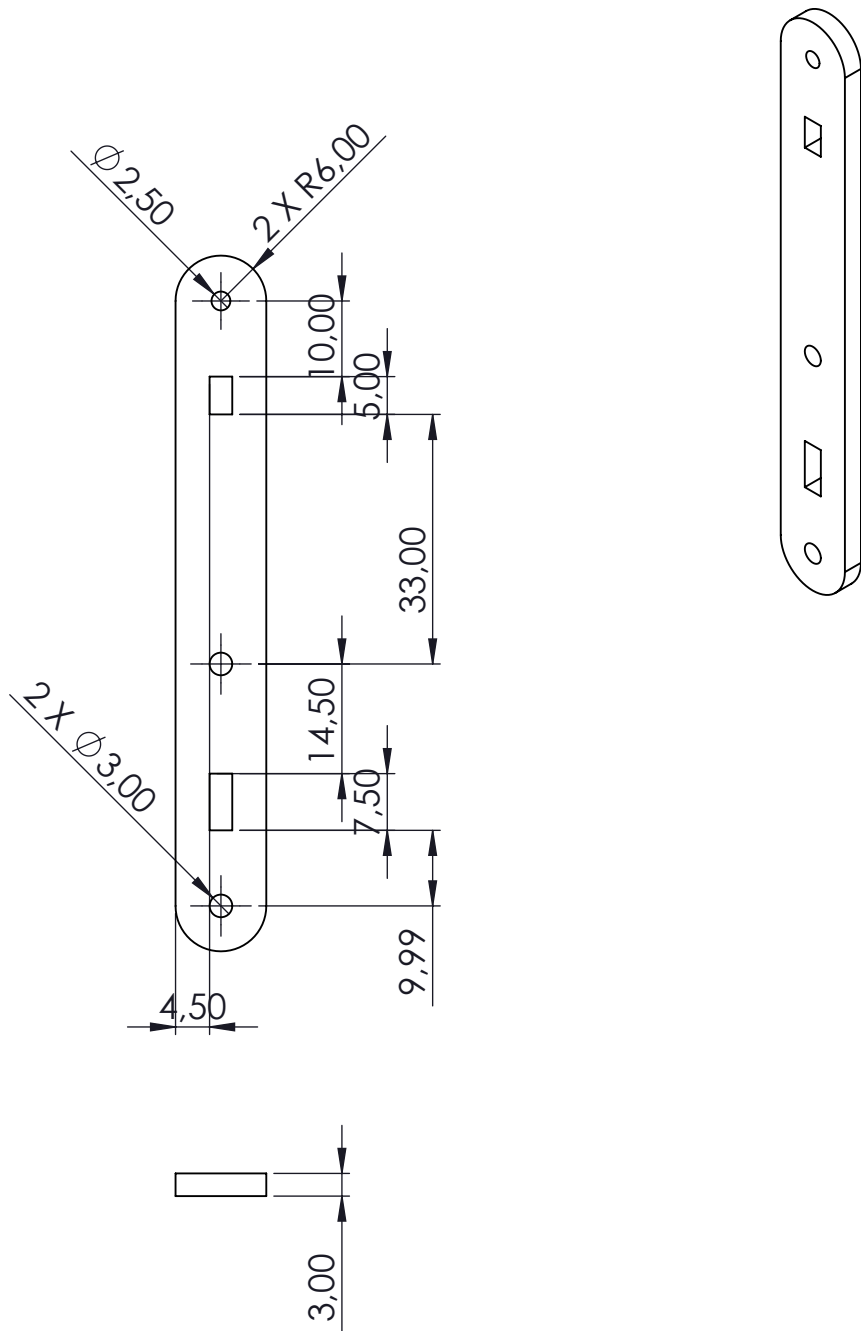
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº12. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>16</p>






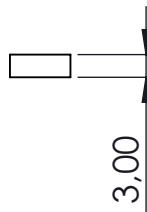
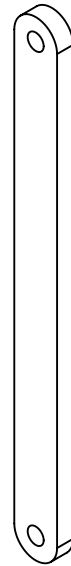
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº13. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano N°</p> <p>17</p>

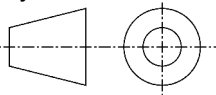




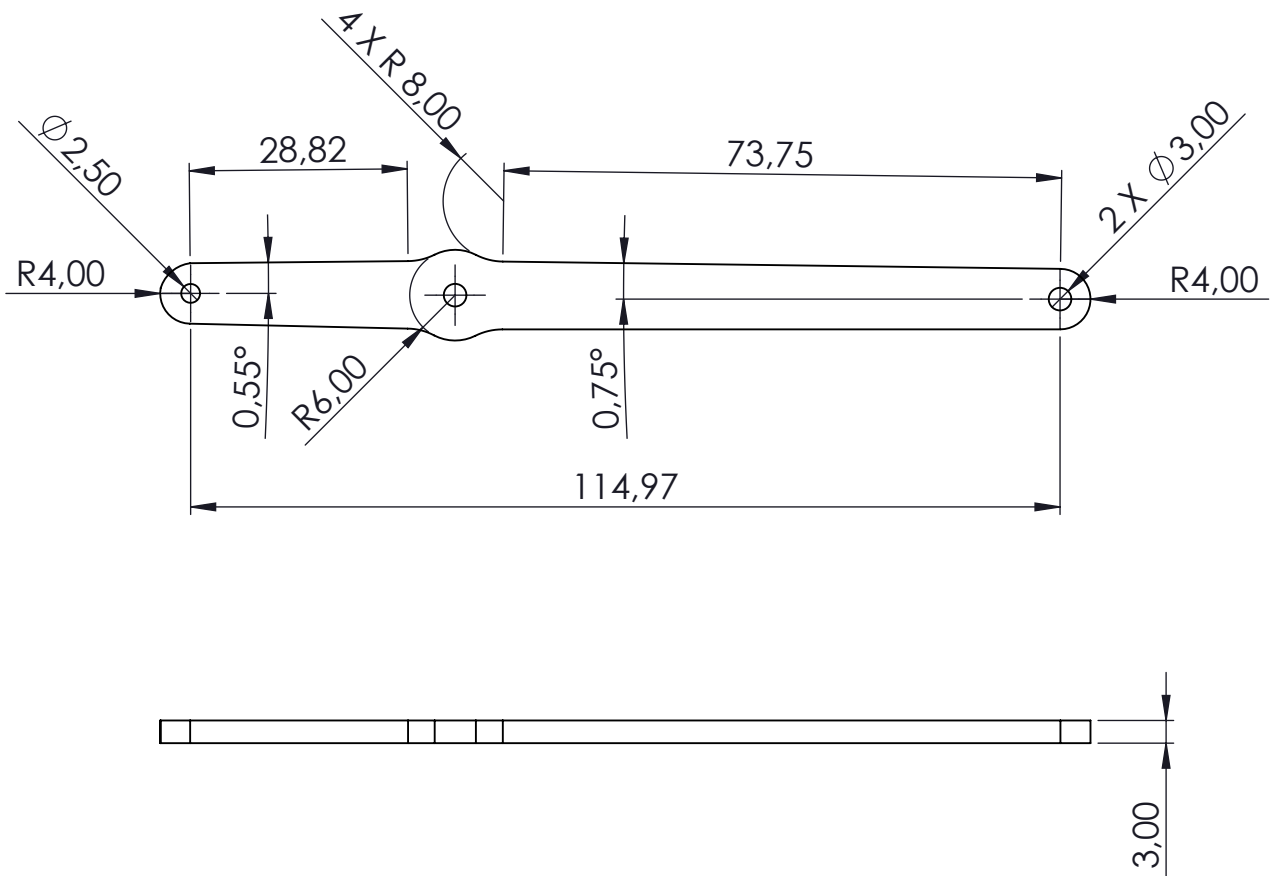
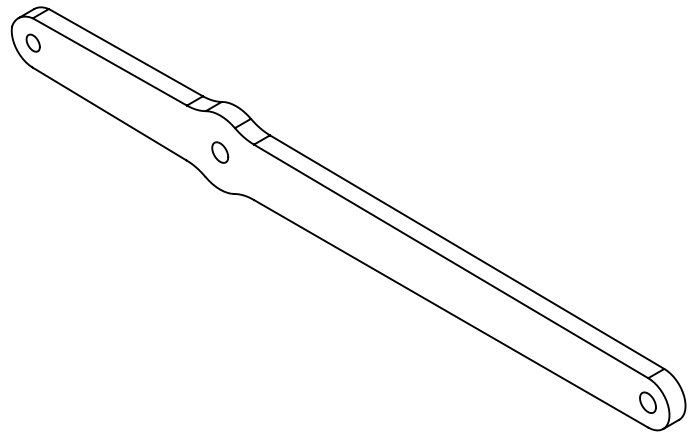
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº14. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>18</p>



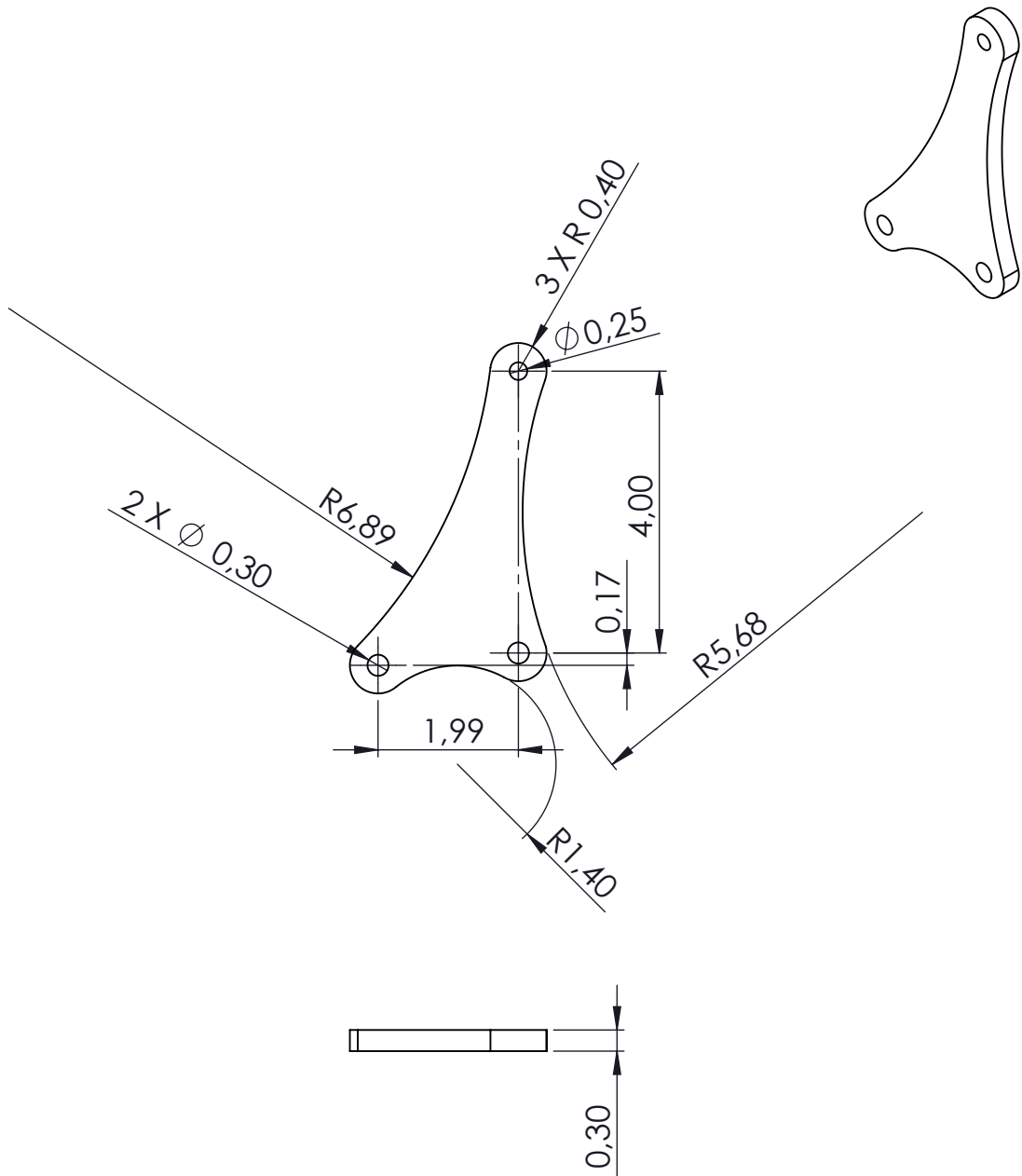
Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 1:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza del brazo manipulador robótico nº15. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano Nº 19

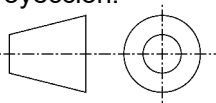




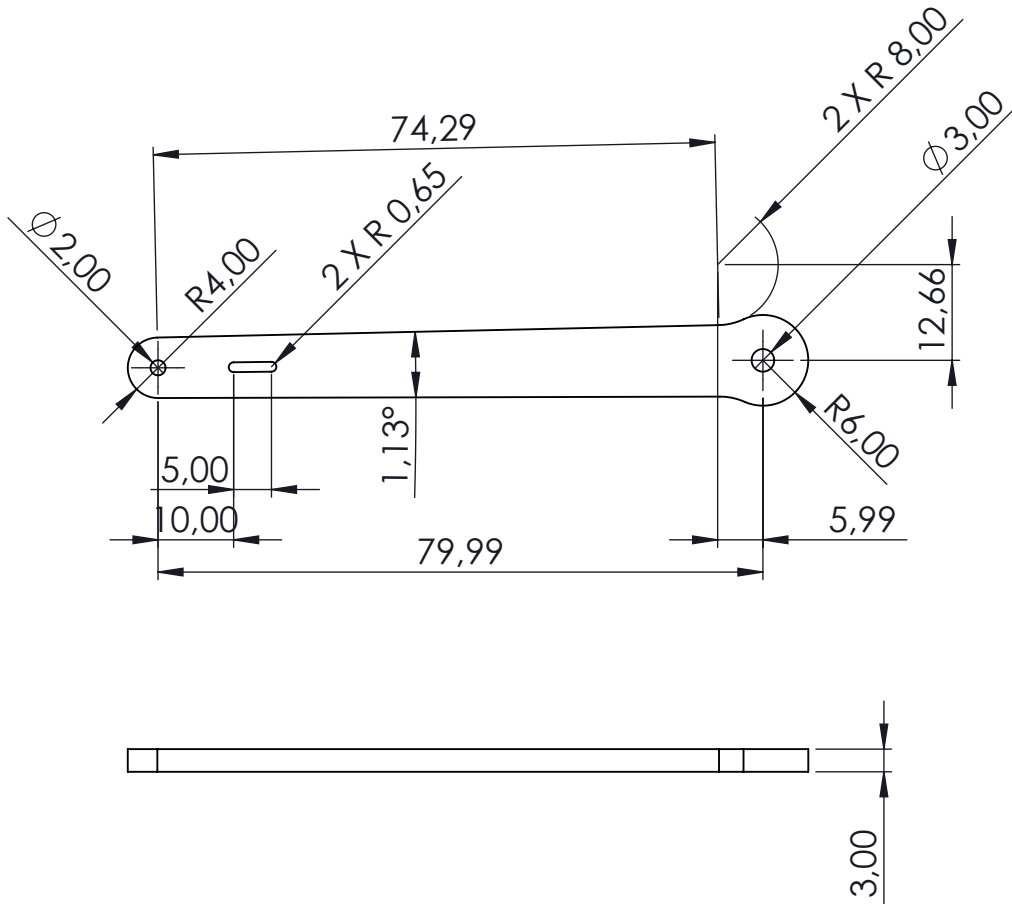
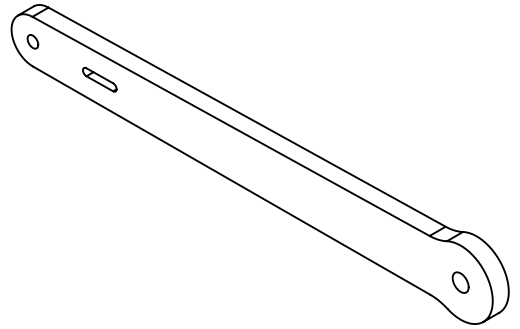
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº16. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>20</p>

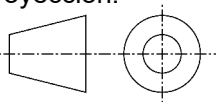




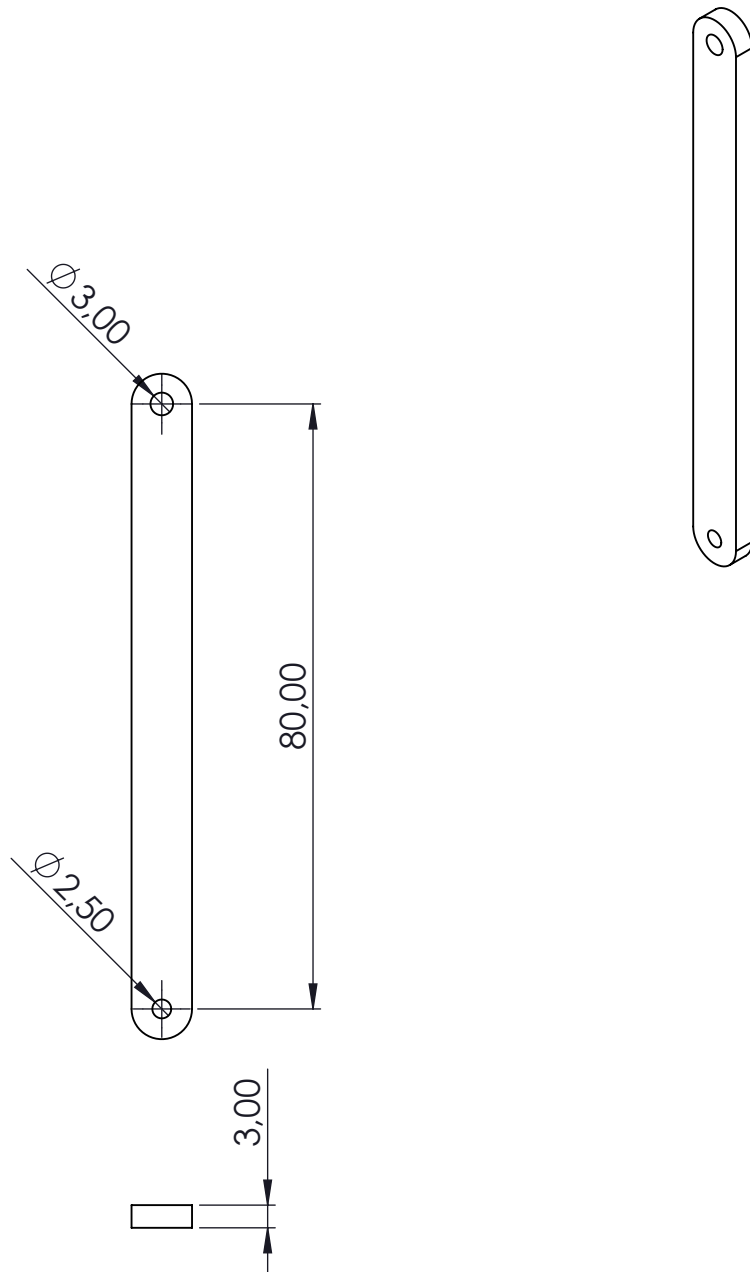
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº17. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano N°</p> <p>21</p>

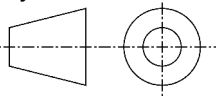




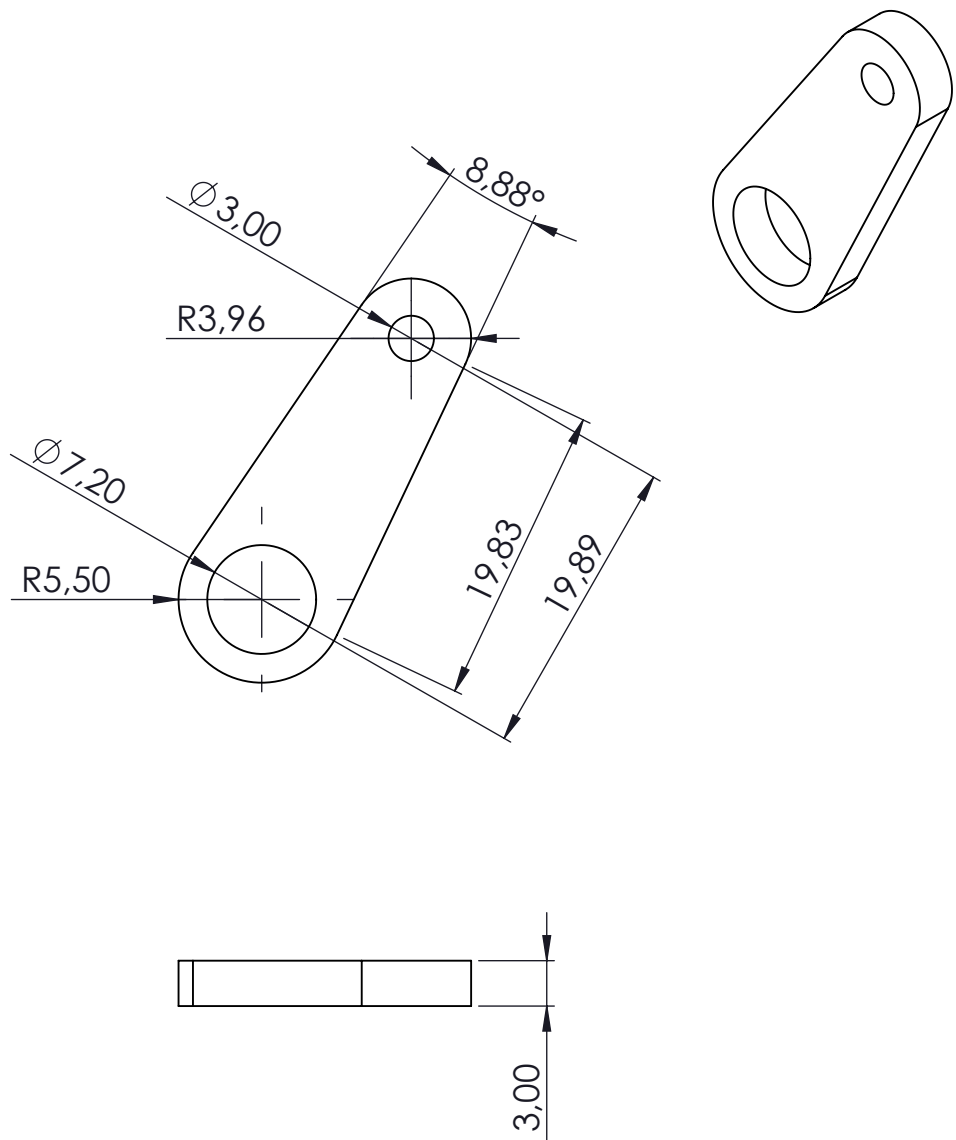
Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 1:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza del brazo manipulador robótico nº18. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano Nº 22

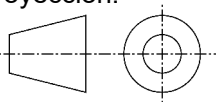




Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 1:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza del brazo manipulador robótico nº19. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano Nº 23

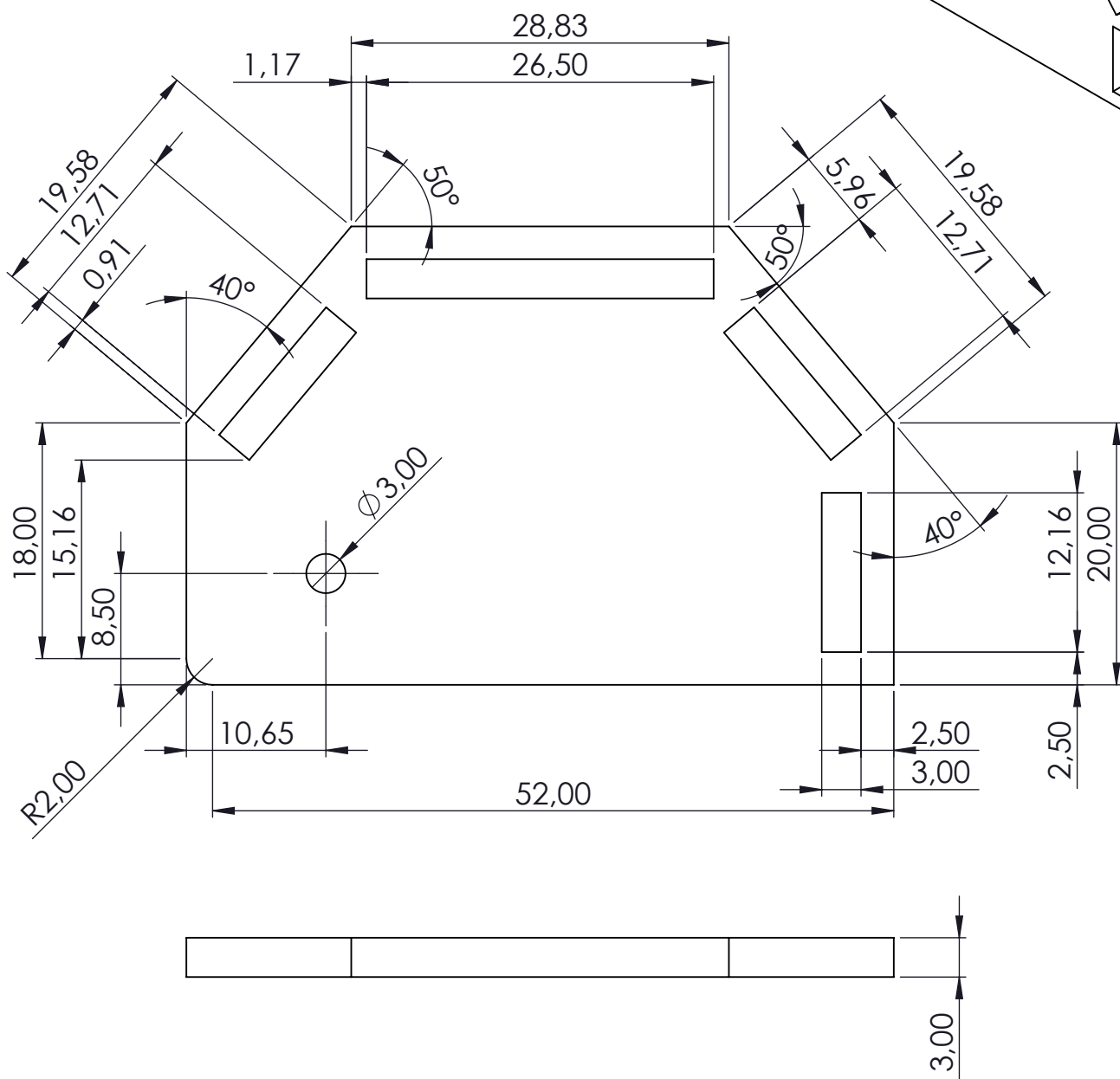
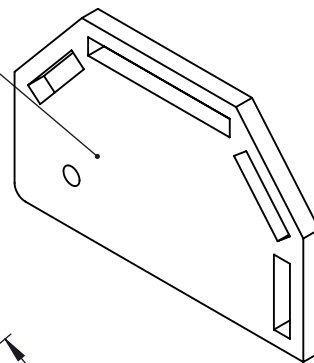


<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº20. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano N°</p> <p>24</p>

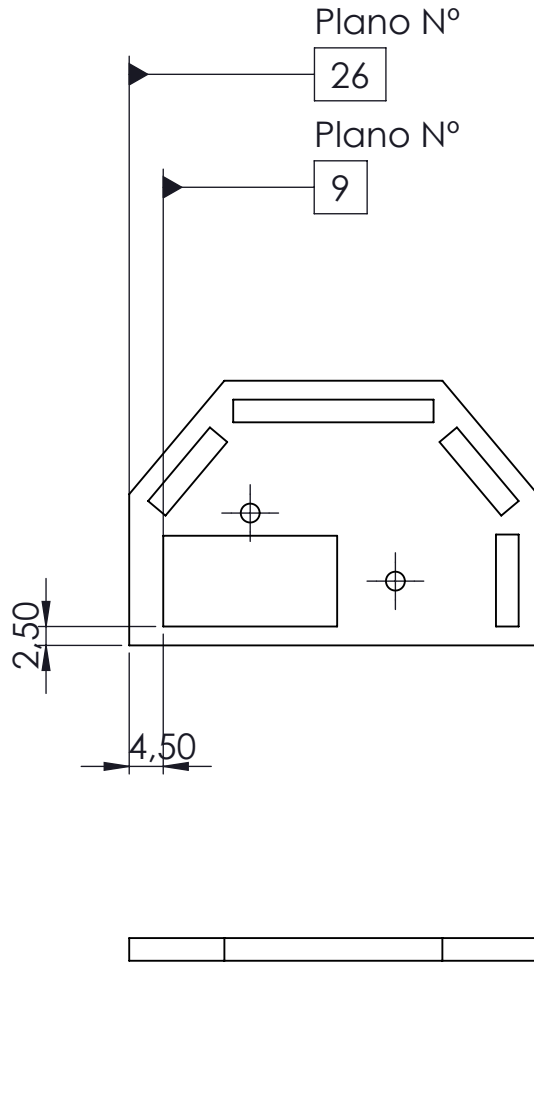
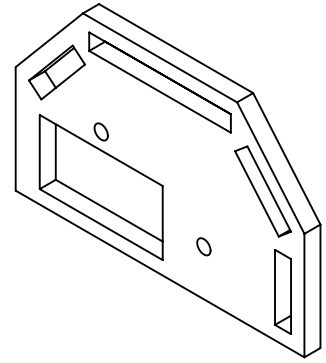


Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 5:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza del brazo manipulador robótico nº21. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano Nº 25

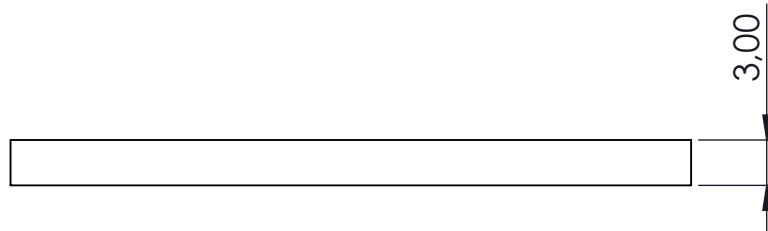
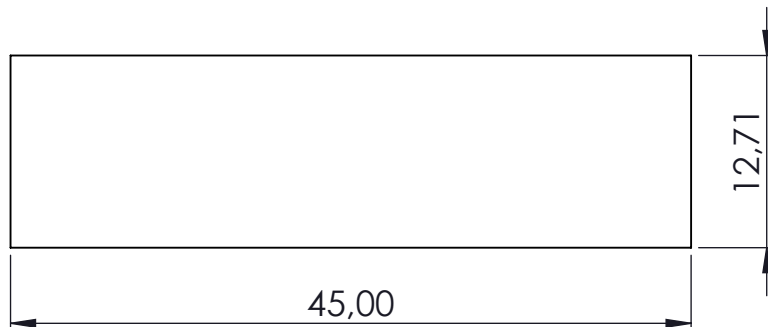
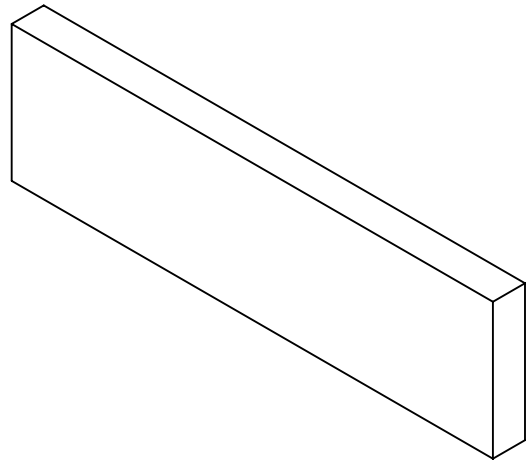
Vista isométrica 1:1

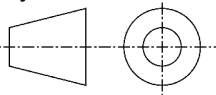




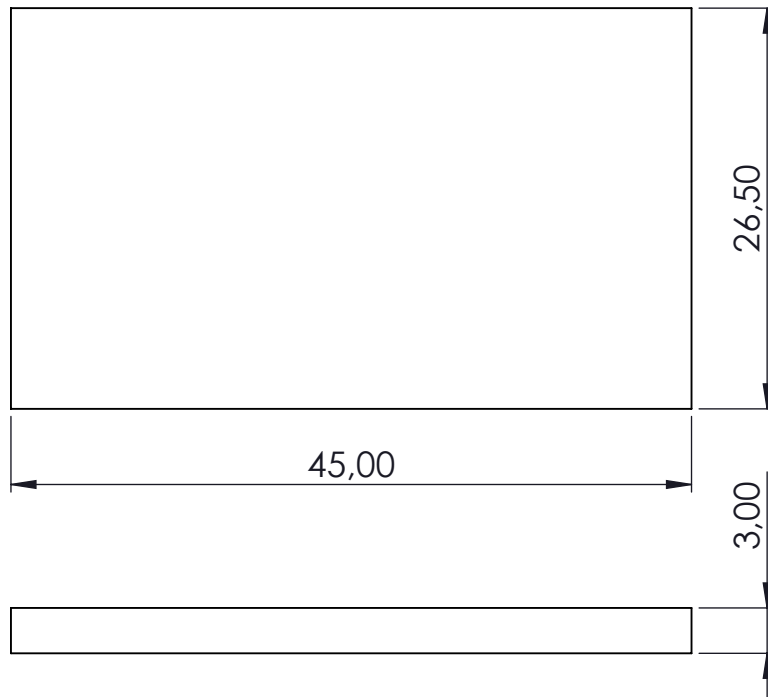
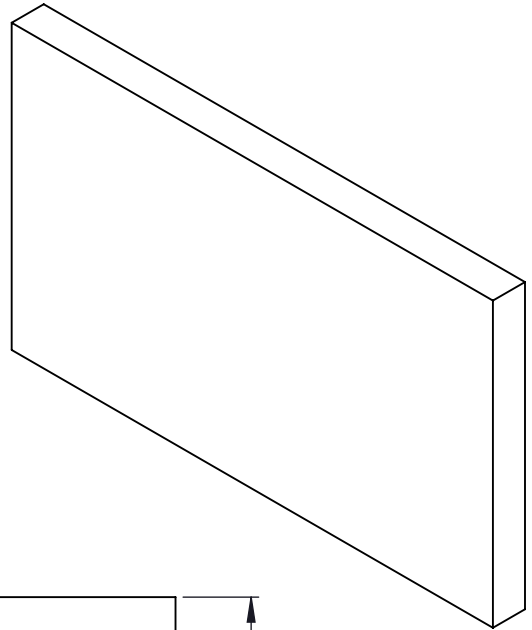
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>	<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>	<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p> <p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº22. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano N°</p> <p>26</p>

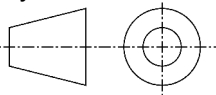




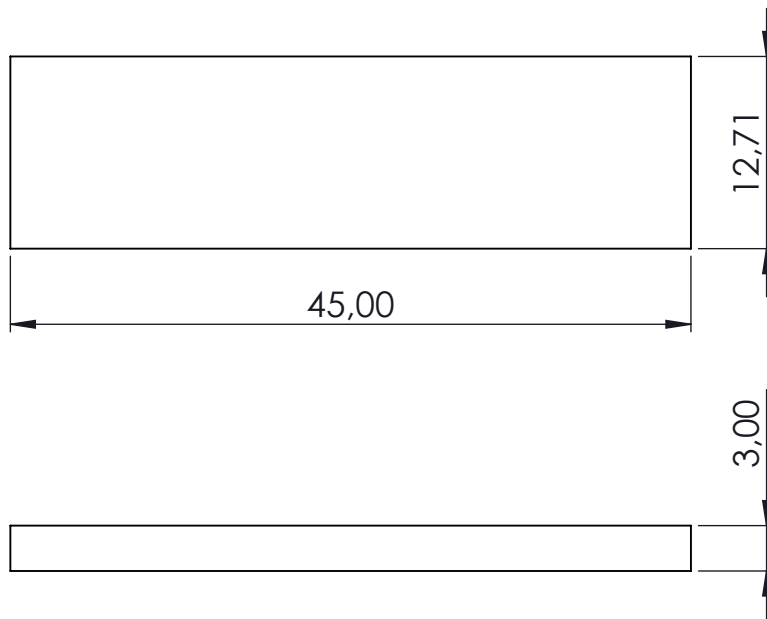
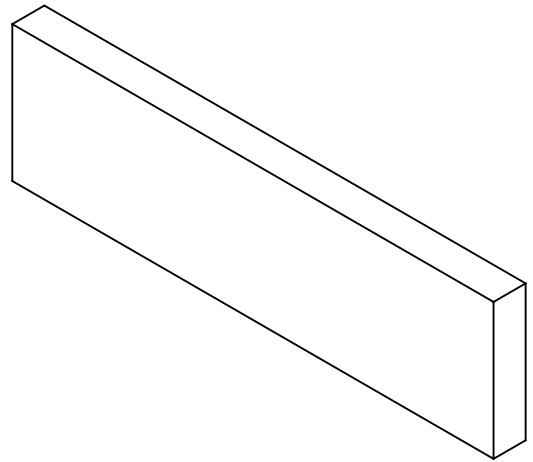
<p>Proyección:</p>	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p> <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>1:1</p>
<p>Universidad:</p> <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº23. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano N°</p> <p>27</p>

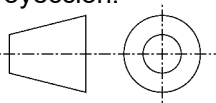




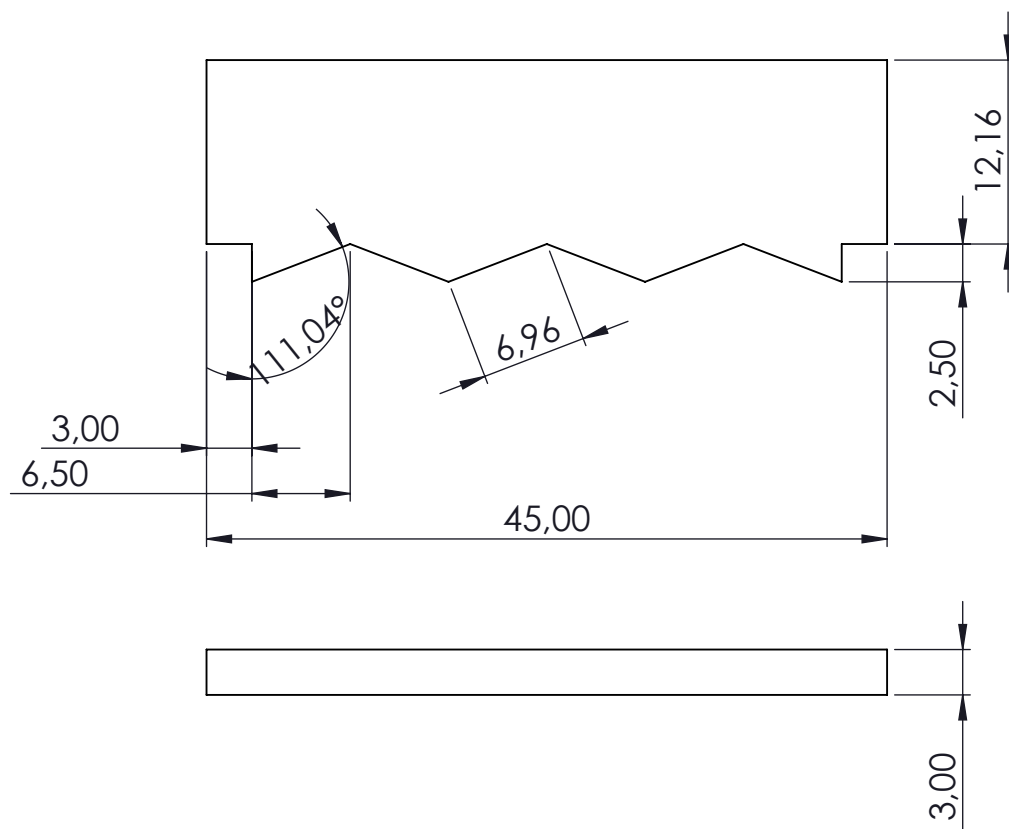
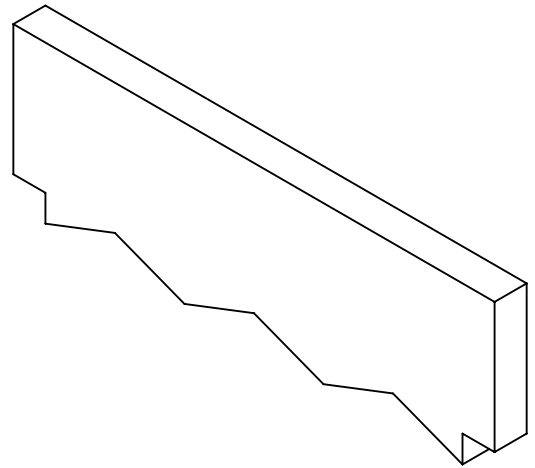
Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE- EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 2:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Estesó Formato: A4	TÍTULO: Pieza del brazo manipulador robótico nº24. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano Nº 28

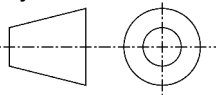




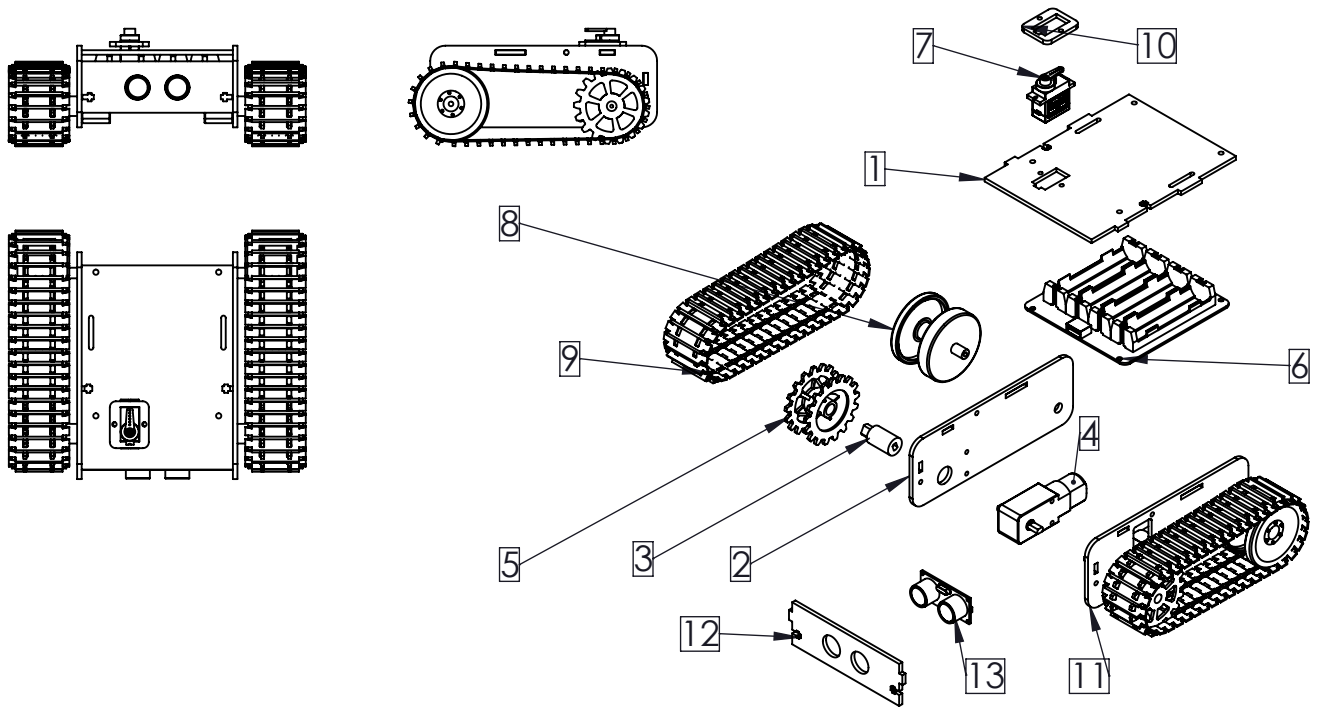
<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº25. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>29</p>



<p>Proyección:</p> 	<p>PROYECTO:</p> <p>Modelado, electrónica y programación de robot excavadora</p>		<p>Material:</p> <p>Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015</p>
<p>Escuela:</p>  <p>Escuela Técnica Superior de Ingeniería del Diseño</p>	<p>PLANO:</p> <p>Dibujo de subconjunto</p> <p>HOJA: 1/1</p>		<p>Fecha: 25/06/21</p> <p>Escala:</p> <p>2:1</p>
<p>Universidad:</p>  <p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Autor: David Martínez Esteso</p> <p>Formato: A4</p>	<p>TÍTULO:</p> <p>Pieza del brazo manipulador robótico nº26. Rediseño del modelo del tutor Leopoldo Armesto Ángel</p>	<p>Plano Nº</p> <p>30</p>

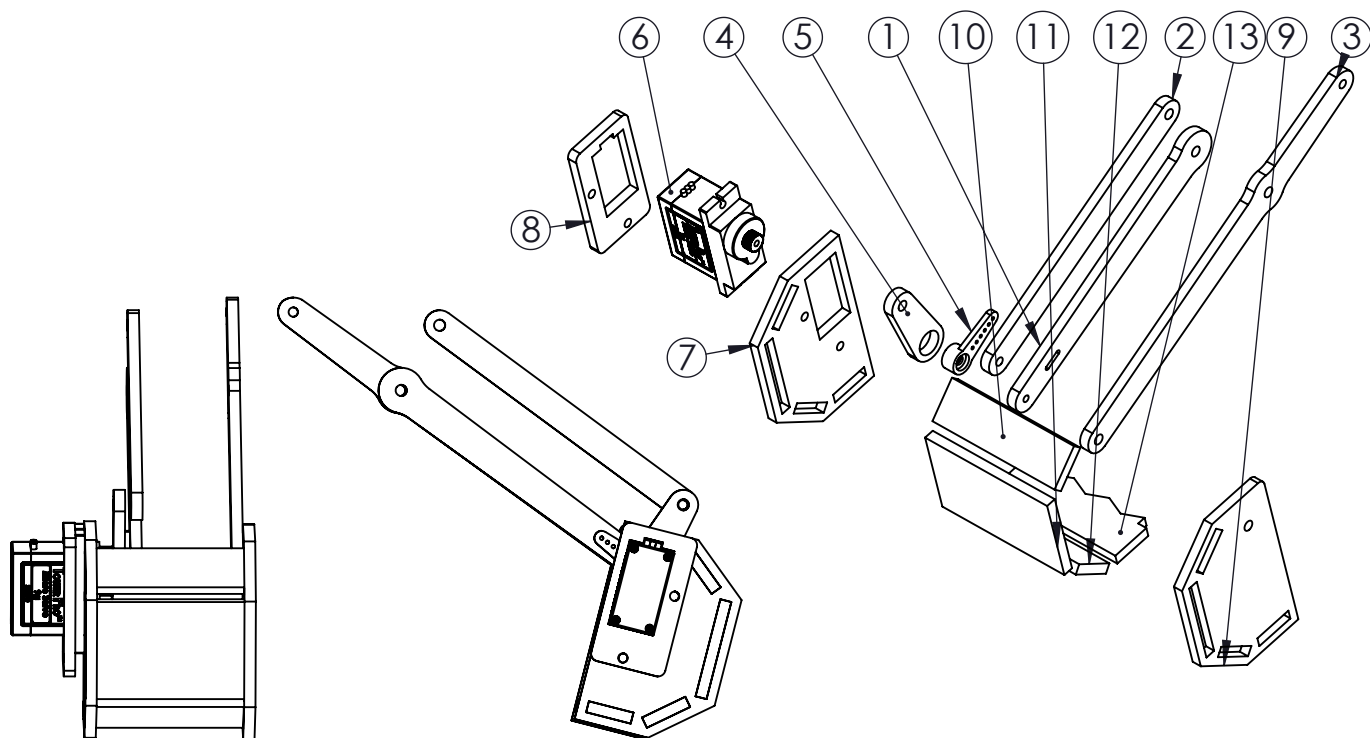


Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		Material: Contrachapado de madera 3 mm UNE-EN636:2012+A1:2015
Escuela:  Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Dibujo de subconjunto HOJA: 1/1		Fecha: 25/06/21 Escala: 2:1
Universidad:  UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Pieza del brazo manipulador robótico nº27. Rediseño del modelo del tutor Leopoldo Armesto Ángel	Plano Nº 31



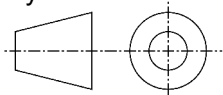
N.º DE ELEMENTO	DESCRIPCIÓN	CANTIDAD	Referencia
1	Base	1	Parte móvil
2	Lateral derecho	1	Parte móvil
3	Pieza de unión	2	Pack adquirido
4	Motor DC	2	Pack adquirido
5	Rueda motriz	2	Pack adquirido
6	Soporte pilas 18650	1	Adquirido A1
7	Servomotor SG90	1	Adquirido A2
8	Rueda conducida	2	Pack adquirido
9	Oruga	2	Pack adquirido
10	Pieza nº10	1	Manipulador
11	Lateral izquierdo	1	Parte móvil
12	Pieza frontal	1	Parte móvil
13	Sensor HC-SR04	1	Adquirido A3

Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora	
Escuela: Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Diagrama de montaje HOJA: 1/1	Fecha: 25/06/21 Escala: 1:5
Universidad: UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Ensamblaje de la parte móvil Plano N° 32



N.º DE ELEMENTO	DESCRIPCIÓN	CANTIDAD	Referencia
1	Pieza nº19	1	Manipulador
2	Pieza nº20	1	Manipulador
3	Pieza nº17	1	Manipulador
4	Pieza nº21	1	Manipulador
5	Manilla SG90	1	Adquirido A2
6	Servo SG90	1	Adquirido A2
7	Pieza nº23	1	Manipulador
8	Pieza nº5	1	Manipulador
9	Pieza nº22	1	Manipulador
10	Pieza nº24	1	Manipulador
11	Pieza nº25	1	Manipulador
12	Pieza nº26	1	Manipulador
13	Pieza nº27	1	Manipulador

Proyección:



PROYECTO:

Modelado, electrónica y programación de robot excavadora

Escuela:



PLANO:

Diagrama de montaje

HOJA: 1/1

Fecha: 25/06/21

Escala:
1:2

Universidad:



Autor: David Martínez Esteso

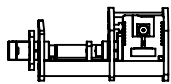
Formato: A4

TÍTULO:

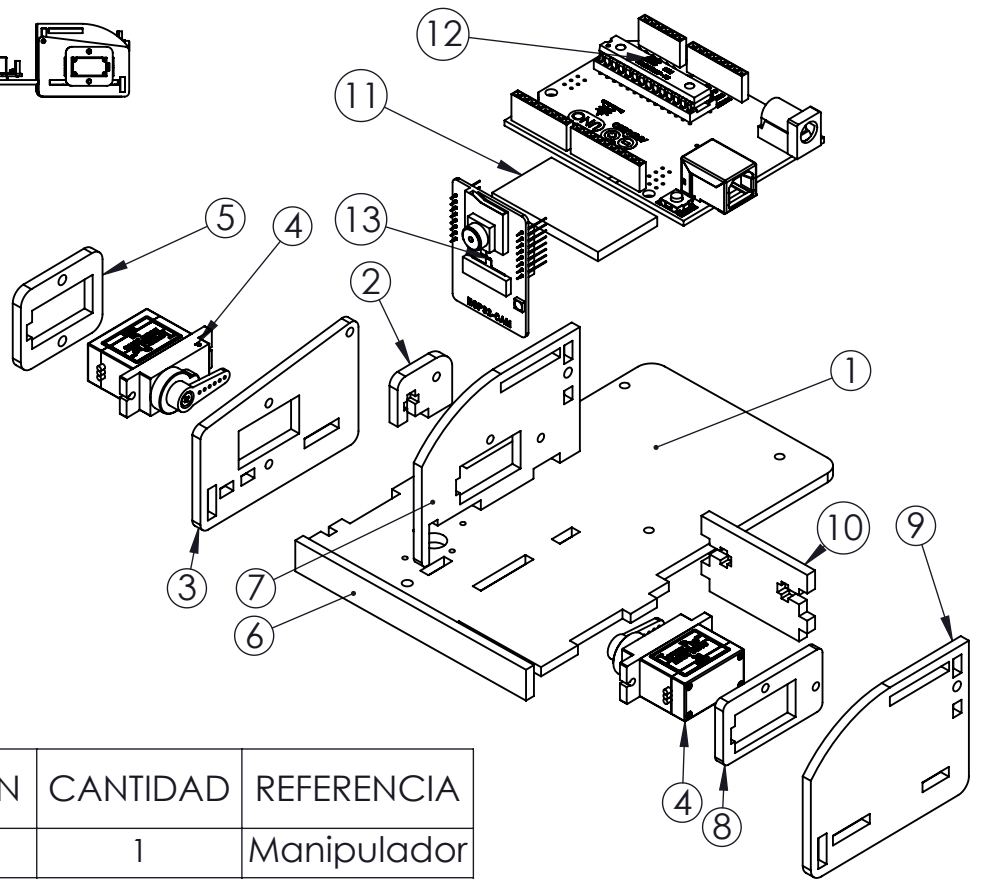
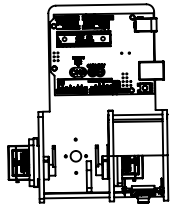
Ensamblaje de la pala del manipulador

Plano Nº

33



Escala 1:5



N.º DE ELEMENTO	DESCRIPCIÓN	CANTIDAD	REFERENCIA
1	Pieza nº1	1	Manipulador
2	Pieza nº4	1	Manipulador
3	Pieza nº2	1	Manipulador
4	Servo SG90	2	Adquirido A2
5	Pieza nº10	1	Manipulador
6	Pieza nº7	1	Manipulador
7	Pieza nº3	1	Manipulador
8	Pieza nº5	1	Manipulador
9	Pieza nº6	1	Manipulador
10	Pieza nº8	1	Manipulador
11	Pieza nº9	1	Manipulador
12	Arduino UNO	1	Adquirido A4
13	ESP32-CAM	1	Adquirido A5

Proyección: 	PROYECTO: Modelado, electrónica y programación de robot excavadora		
Escuela: Escuela Técnica Superior de Ingeniería del Diseño	PLANO: Diagrama de montaje HOJA: 1/1		Fecha: 25/06/21
			Escala: 1:2
Universidad: UNIVERSITAT POLITÈCNICA DE VALÈNCIA	Autor: David Martínez Esteso Formato: A4	TÍTULO: Ensamblaje de la cabina del manipulador	Plano N° 34



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Documento nº3:
Pliego de
condiciones

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Esteso

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

Contenido

1. Objetivo y alcance del pliego	2
2. Condiciones y normas de carácter general	2
3. Especificaciones técnicas	2
3.1. Objeto	2
3.2. Condiciones de los materiales	2
3.3. Condiciones de la ejecución	3
3.4. Prueba de servicio	3
4. Especificaciones económicas	4
5. Entrega	4

1. Objetivo y alcance del pliego

El presente pliego de condiciones tiene como objetivo definir las extensiones del contrato entre el promotor, Leopoldo Armesto Ángel, y el contratista, David Martínez Estesos.

En este proyecto se propone el modelado, selección de unos componentes electrónicos y programación, a cargo del contratista.

Se tratan para este proyecto las condiciones generales, técnicas, económicas y legales que se han de tener en cuenta.

2. Condiciones y normas de carácter general

Con carácter general, se llevará a cabo la realización de este proyecto con todas las instrucciones y consideraciones aportadas en los documentos de la memoria y los planos.

Tanto promotor como proyectista se comprometen a cumplir las condiciones de este pliego hasta el fin del contrato, antes de la fecha límite estipulada para el 15 de julio de 2021 desde el momento de la firma en marzo de 2021.

3. Especificaciones técnicas

3.1. Objeto

El objetivo de esta especificación técnica es definir las condiciones técnicas mínimas exigibles que se deben llevar a cabo para la correcta realización del proyecto.

3.2. Condiciones de los materiales

El material de fabricación del robot excavadora que se desarrolla, especificado en los planos, es contrachapado de madera de 3 mm el cual debe cumplir con la normativa UNE-EN 636:2012+A1:2015. Especialmente, se debe prestar atención al apartado 6.1 de tolerancias puesto que las piezas presentan medidas milimétricas, descartando el punto 8 de utilización en ambientes húmedos.

Optativamente, se podrá realizar la fabricación en material de fibras de densidad media (DM) según UNE-EN 622-5:2010, también de 3 mm de grosor.

Para los tornillos necesarios de métricas M2x6 de rosca y M3x6, M3x12 y M3x25 pasantes, deben cumplir con la normativa UNE-EN ISO 225:2011.

3.3. Condiciones de la ejecución

El método de ejecución del proyecto tiene como principal finalidad la disminución del impacto de posibles errores y corrección temprana de los mismos. El cumplimiento de esto supondrá una reducción tanto en costes materiales y económicos como temporales.

Antes de comenzar con la etapa de modelado, se debe realizar un estudio del funcionamiento y componentes de una máquina excavadora, además de un estudio de la estructura del robot meArm a partir del cual se lleva a cabo la realización de una parte del robot a desarrollar. Durante el proceso de modelado, se realizará el ensamblaje de las piezas diseñadas para verificar la correcta ejecución de estas. El resultado se exportará y preparará para la simulación, se programará estudiando la estructura resultante y finalmente se realizará la construcción y montaje final.

Los equipos informáticos que se utilizan en el proceso de diseño y programación deberán estar homologados en fecha marzo de 2021. Así mismo, el software utilizado debe contar con, al menos, las versiones que se utilizan en este proyecto y se especifican a continuación:

- SolidWorks 2020.
- SolidWorks URDF exporter 1.6.0 (2020)
- V-REP CoppeliaSim 4.1.0.
- Arduino IDE 1.8.13
- MATLAB R2020b

3.4. Prueba de servicio

Se debe comprobar el funcionamiento de cada uno de los componentes electrónicos en pequeños conjuntos o por separado, dentro de las posibilidades. La prueba de servicio para cada uno de los componentes es la siguiente:

Componente electrónico	Verificación	Prueba por realizar	Número de pruebas	Condición de no aceptación
Arduino UNO	Gestión de señales de entrada y salida	Programa sencillo en Arduino IDE (ejemplo: iluminación de LEDs con bread board)	1	Incorrecto funcionamiento de más del 5% de los pines.
Placa de expansión con protoshield Arduino UNO	Conexiones con Arduino UNO + bread board	Junto con Arduino UNO, programa sencillo en Arduino IDE (ejemplo: iluminación de LEDs)	1	Errores en las conexiones con Arduino UNO o bread board defectuosa

Motor DC + L293D	Giro de eje en ambos sentidos y parada.	Control en ambos sentidos de motor DC mediante componente L293D en Arduino IDE.	1	Sentidos de giro y parada incorrectos.
Servomotor SG90/MG90S	Posicionamiento angular.	Indicación de un ángulo concreto especificado mediante código en Arduino IDE.	1	Funcionamiento defectuoso o ángulo ejecutado incorrectamente.
Sensor de ultrasonidos HC-SR04	Medición de distancias.	Medición de distancia de objetos mediante código en Arduino IDE.	1	Incorrecto funcionamiento o distancias obtenidas irreales.
ESP32-CAM	Monitorización mediante cámara a través de WiFi.	Programación mediante Arduino UNO o módulo USB a TTL.	1	No muestra la imagen o es defectuosa.
Pila 18650	3.7 V aproximados entre sus bornes.	Medición con multímetro.	1	No hay 3.7 V aproximados en sus bornes.
Soporte de pilas 18650	Proporciona 5V en las salidas indicadas.	Medición con multímetro.	1	No hay 5V aproximados en los pines de salida que especifica esta tensión.

4. Especificaciones económicas

Los pagos se realizarán en 2 partidas. Se debe aportar el 20 % del total del presupuesto del proyecto especificado en el mismo, en el momento de firma del contrato. Esto servirá también a modo de fianza. El 80 % restante se aportará en la entrega del proyecto, siempre y cuando se cumpla con el plazo establecido.

En caso de no cumplir con los plazos establecidos, se descontará un 4% del pago sobre el total por día de retraso al promotor a modo de indemnización.

5. Entrega

Para la entrega de este proyecto, se aportarán imágenes y/o vídeos del resultado final del montaje y escenas llevadas a cabo, además de los documentos mínimos exigibles de un proyecto técnico.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

MODELADO, ELECTRÓNICA Y PROGRAMACIÓN DE ROBOT EXCAVADORA

Documento nº4:
Presupuesto

TRABAJO FINAL DEL

Grado en Ingeniería Electrónica Industrial y Automática

REALIZADO POR

David Martínez Esteso

TUTORIZADO POR

Leopoldo Armesto Ángel

CURSO ACADÉMICO: 2020/2021

Contenido

1. Presupuesto por costes según naturaleza	2
--	---

1. Presupuesto por costes según naturaleza

Ref	Ud	Descripción	Precio	Cantidad	Total
Materiales					
m1	ud	Arduino UNO con cable de conexión USB	9,99 €	1	9,99 €
m2	ud	Placa de expansión con protoshield Arduino UNO	6,38 €	1	6,38 €
m3	ud	Pack: Motor DC con motorreductor amarillo + Cinta de oruga + ruedas+tornillería	5,11 €	2	10,22 €
m4	ud	Sensor de ultrasonidos HC-SR04	1,92 €	1	1,92 €
m5	ud	Receptor con sensor de IR para Arduino	0,76 €	1	0,76 €
m6	ud	Mando para control IR	1,46 €	1	1,46 €
m7	ud	Servomotor SG90	2,02 €	4	8,08 €
m8	ud	ESP32-CAM	6,41 €	1	6,41 €
m9	ud	Soporte de pilas 18650	6,54 €	1	6,54 €
m10	ud	Pack: 4 Pilas 18650	11,99 €	1	11,99 €
m11	ud	Piezas de corte láser en contrachapado de madera 3 mm	42,35 €	1	42,35 €
m12	ud	Cables 120 piezas macho-hembra, macho-macho, hembra-hembra Arduino	6,99 €	1	6,99 €
m13	ud	Tornillos de rosca M2x6	0,99 €	1	0,99 €
m14	ud	Tornillos pasantes M3x6, M3x12, M3x25, M3x30	3,96 €	1	3,96 €
Licencias					
l1	ud	Licencia SolidWorks 3 meses	1.010,00 €	1	1.010,00 €
l2	ud	MATLAB 1 año	800,00 €	1	800,00 €
l3	ud	Arduino IDE	- €	1	0,00 €
l4	ud	V-REP CoppeliaSim	- €	1	0,00 €
Mano de obra					
h1	h	Servicio de ingeniero cualificado	14,88 €	300	4.464,00 €
Medios auxiliares					
	%	Medios auxiliares sobre costes directos	5%	6.392,04 €	319,60 €
SUBTOTAL					6.711,64 €
			IVA	21%	
PRESUPUESTO TOTAL					8.121,09 €