

Francisco José Del Moral Conde

Diseño de prototipo un robot industrial
y su control mediante una interfaz
gráfica de usuario



UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Jaén

CONTROL DE UN PROTOTIPO DE ROBOT INDUSTRIAL CON UNA GUI DISEÑADA EN APPDESIGNER (MATLAB)

Alumno: Francisco José Del moral Conde

Tutor: Elisabet Estévez Estévez

Dpto.: Ing. Electrónica y automática

Febrero, 2021

Francisco José Del Moral Conde

Diseño de prototipo un robot industrial
y su control mediante una interfaz
gráfica de usuario

Grado en Ingeniería Electrónica Industrial



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ing. Electrónica y automática

Dña. **Elisabet Estévez Estévez**, tutor del Proyecto Fin de Carrera titulado **Control de un prototipo de robot industrial con una GUI diseñada en AppDesigner (MATLAB)**, que presenta **Francisco José Del Moral Conde**, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Febrero de 2021

El alumno:

Jaén, a 18 de febrero de 2021

Una firma manuscrita en tinta negra que parece decir "Fco. José" seguida de un nombre de pila y un apellido, todo en una sola línea fluida.

Francisco José Del Moral Conde

Los tutores:

Elisabet Estévez Estévez

Resumen

Este trabajo tiene por objetivo la realización del diseño de un prototipo de robot antropomórfico y el control del mismo mediante una interfaz gráfica de usuario creada en Matlab, dicho robot está basado en un modelo infradimensionado del modelo KR16 del fabricante KUKA.

Como se ha mencionado, el robot, llamado *Franuc 24*, parte del diseño de un robot del fabricante KUKA, mecánicamente no tiene las mismas prestaciones que un robot industrial en lo que se refiere a precisión, debido a que *Franuc 24* es movido por motores paso a paso con una precisión de 1.8° sin sistemas de engranaje o poleas en todas las articulaciones, lo que permitiría afinar la precisión del robot, simplemente se acoplan los motores a los ejes de rotación.

Una vez terminado el diseño del robot, se ha pasado a diseñar la interfaz gráfica de usuario (GUI) para el control del mismo, empezando por la creación de un gemelo digital (DT) el cual ayudará a visualizar los movimientos y permitirá testear cualquier programa antes de que lo haga el robot físico. Una vez creada esta realidad virtual, se ha calculado la cinemática directa mediante el algoritmo de Denavit-Hartenberg y la cinemática inversa mediante métodos geométricos. Por último, se han establecido los límites posicionales de nuestro robot, que vienen dados por limitaciones geométricas de nuestro diseño.

Abstract

The objective of this project is to design an anthropomorphical robot prototype and control it through a graphical user interface created in Matlab, said robot is based on an undersized model of the KR16 model from the manufacturer KUKA.

As mentioned, the robot, called Franuc 24, part of the design of a robot from the manufacturer KUKA, mechanically does not have the same performance as an industrial robot in terms of precision, because Franuc 24 is driven by stepper motors. by step with a precision of 1.8° without gear systems or pulleys in all the joints, which would allow to improve the precision of the robot, the motors are simply coupled to the rotation axes.

Once the design of the robot is finished, the graphical user interface (GUI) has been designed to control it, starting with the creation of a digital twin (DT) which will help to visualize the movements and allow testing any program before the physical robot does. Once this virtual reality was created, the direct kinematics were calculated using the Denavit-Hartenberg algorithm and the inverse kinematics using geometric methods. Finally, the positional limits of our robot have been established, which are given by geometric limitations of our design.

Índice

Contenido

Resumen.....	III
Abstract.....	IV
Índice	V
Índice de tablas	VII
Índice de figuras.....	VIII
1. MOTIVACIÓN, OBJETIVOS Y ESTRUCTURA	1
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.3 Estructura	2
2 ROBOTS ANTROPOMÓRFICOS.....	3
2.1 Definición de robot antropomórfico	3
2.2 Características de un robot antropomórfico.....	4
2.3 Morfología de un robot antropomórfico	5
2.4 Elementos terminales de un robot.....	5
3 DISEÑO DEL GEMELO DIGITAL (DT).....	7
4. Cinemática del robot.....	9
4.1 Conceptos básicos.....	10
4.1.1 Sistema de coordenadas	10
4.1.2 Matrices de transformación homogénea.....	11
4.2 Introducción al funcionamiento de Franuc 24.....	12
4.3 Cinemática directa del robot	12
4.4 Cinemática inversa del robot.....	17
4.5 Áreas de trabajo del robot	21
5. ENTORNO DE DESARROLLO INTEGRADO PARA FRANUC 24	29
5.1 Fases de la programación.....	29
5.2 Pantalla de INICIO	31
5.3 Pantalla de TAREAS.....	32
5.4 Pantalla de SIMULACIÓN.....	34
5.5 Pantalla de ROBOT	35
6. ROBOT “FRANUC 24”	36
6.1 Diseño imprimible del robot.	36

6.1.1 Base	36
6.1.2 Brazo inferior	37
6.1.3 Brazo superior	37
6.1.4 Muñeca y pinza	38
6.2. Componentes hardware	39
6.3 Software	43
7. MANUAL DE USUARIO	44
8. PRESUPUESTO	48
9. CONCLUSIONES Y TRABAJOS FUTUROS.....	50
ANEXOS	54
ANEXO I: DATASHEETS	54
.....	57
.....	57
ANEXO II: PLANOS DISEÑO DE PIEZAS	58
.....	58
.....	59
Bibliografía	67

Índice de tablas

Tabla 1: Algunas de las operaciones que pueden realizar los robots industriales.....	10
Tabla 2: Parámetros Denavit-Hartenberg.....	16
Tabla 3: Parámetros del robot.....	17
Tabla 4: Variables para el cálculo de los límites.....	18
Tabla 5: Presupuesto del material necesario para el desarrollo del robot.....	48
Tabla 6: Presupuesto del material necesario para la elaboración del proyecto.....	49
Tabla 7: Presupuesto de las licencias de software.....	49
Tabla 8: Presupuesto final.....	50

Índice de figuras

Figura 1: Robots antropomórficos (KUKA, FANUC, UNIVERSAL ROBOTS, ABB).....	3
Figura 2: Área de trabajo del robot KUKA KR6.....	4
Figura 3: Similitud entre brazo y robot antropomórfico.....	5
Figura 4: Pinza por accionamiento eléctrico y elemento final de ventosa.....	6
Figura 5: Partes del robot “FRANUC 24”.....	7
Figura 6: Creación del archivo .wrl.....	8
Figura 7: Antes vs después tras la configuración de la realidad virtual.....	9
Figura 8: configuración de la realidad virtual y creación de nodos.....	9
Figura 9: TCP de un robot.....	11
Figura 10: Robot “Franuc 24” y sus articulaciones.....	13
Figura 11: Parámetros Denavit-Hartenberg para un eslabón giratorio.....	16
Figura 12: Estructura del robot Franuc 24.....	17
Figura 13: Diagrama de relación entre cinemática directa e inversa.....	18
Figura 14: Teorema del coseno.....	19
Figura 15: Estructura brazo del robot Franuc 24.....	19
Figura 16: Teorema del seno.....	20
Figura 17: Estructura del robot.....	20
Figura 18: Áreas de trabajo en el plano XZ.....	22
Figura 19: Posición del robot en las dos áreas.....	22
Figura 20: Soluciones codo arriba y codo abajo.....	23
Figura 21: Ángulo QPxmin.....	28

Figura 22: Pantalla de inicio (con usuario aún sin acceder).....	32
Figura 23: Pantalla de tareas.....	32
Figura 24: Pantalla de SIMULACIÓN.....	34
Figura 25: ROBOTsim.....	35
Figura 26: Pantalla de ROBOT.....	35
Figura 27: Mecanismo base.....	36
Figura 28: Mecanismo brazo inferior.....	37
Figura 29: Mecanismo brazo superior.....	37
Figura 30: Mecanismo situado en la muñeca.....	38
Figura 31: Mecanismo de la pinza.....	38
Figura 32: Control de potencia del robot "Franuc 24".....	39
Figura 33: Conexionado DRV8825.....	39
Figura 34: Arduino UNO R3.....	40
Figura 35: Driver DRV8825.....	40
Figura 36: Fuente de alimentación.....	41
Figura 37: Condensador para filtrado.....	41
Figura 38: Motor paso a paso NEMA 17.....	42
Figura 39: Servomotor SG90.....	42
Figura 40: Paquete de Matlab para arduino.....	43
Figura 41: Pantalla de Inicio.....	44
Figura 42: Pantalla de Tareas.....	45
Figura 43: Programación de tareas.....	45
Figura 44: Pantalla de simulación.....	46
Figura 45: Pantalla de ROBOT.....	47

1. MOTIVACIÓN, OBJETIVOS Y ESTRUCTURA

En este capítulo se presenta la estructura de dicho trabajo, la motivación y los objetivos a cumplir.

1.1 Motivación

Observando alrededor, se puede ver como la robótica es algo imprescindible en la industria, la primera patente fue registrada en 1956 a manos de George Devol con el primer manipulador programable y poco a poco se ha ido avanzando hasta el nivel de poder realizar procedimientos complejos en distintos sectores, tanto industrial, militar, farmacéutico, etc...

Se puede deducir, que la robótica está en auge, demandando que el personal encargado de estos procesos (ingenieros y técnicos) lleguen con una buena base de conocimientos sobre el manejo de estas tecnologías.

La robótica es una ciencia multidisciplinar que se encarga del estudio de la programación y mecánica de los robots, el adjetivo de multidisciplinar es debido a que engloba campos tan variados como la informática, la electrónica, el álgebra y la mecánica, entre otras disciplinas. El robot construido en este proyecto es un robot antropomórfico diseñado con el objetivo de acercar a los alumnos de distintas áreas los conocimientos necesarios para la creación de un robot desde cero, empezando por su realidad virtual, la cual es interesante en el caso de que el alumnado no disponga del material necesario para el ensamblaje de un robot, posteriormente, el diseño de su controlador y, si es posible, la puesta en marcha del prototipo impreso en 3D junto con su control de potencia.

La robótica tiene dos formas diferentes para ser explicada, la primera, trata sobre el diseño del robot, tanto mecánico, cinemático, dinámico e informático, y la segunda, trata sobre sus usos y programación de sus movimientos, estas explicaciones serán desarrolladas a lo largo de este proyecto.

Por último, el proyecto concluirá dándole “vida” al robot “*Franuc 24*” mediante la impresión del prototipo en PLA y ensamblándolo junto a su control de potencia (motores y electrónica necesaria).

1.2 Objetivos

La titulación del Grado en Ingeniería Electrónica Industrial, como todas las titulaciones impartidas en la Escuela Politécnica Superior de Jaén, deben de concluir según la normativa del plan vigente con la elaboración y la defensa por parte del estudiante.

En lo que a docencia se refiere, este proyecto tiene como objetivo principal que el alumnado pueda crear su propio robot y, gracias a su gemelo digital no sería necesario tener físicamente un robot para controlarlo. Lo anteriormente dicho, ayudaría en una situación como la que se está experimentando desde marzo de 2020 y que previsiblemente, a fecha de esta defensa, aún se está viviendo.

1.3 Estructura

En este proyecto se puede diferenciar la siguiente estructura:

- Diseño del robot “*Franuc 24*” en Solidworks.
- Creación de un gemelo digital (DT), una realidad virtual del robot mediante Matlab, a través de sus herramientas “Simulink Simscape” y “simulink 3D animation”.
- Realización del estudio de la cinemática directa e inversa, así como la dinámica del robot.
- Creación de una GUI (interfaz gráfica de usuario) mediante AppDesigner de Matlab.
- Creación de la interfaz Matlab-Arduino para el control de los motores paso a paso.

- Diseño del control de potencia y una versión imprimible del robot “Franuc 24”.
- Elaboración de un protocolo de utilización adecuado del robot.
- Puesta en marcha del robot.

2 ROBOTS ANTROPOMÓRFICOS

En este capítulo se realizará una introducción a los robots antropomórficos, se finalizará mostrando el diseño del robot *Franuc 24*, para ello, se ha utilizado *Solidworks*. Como se ha comentado en el resumen, el diseño parte del robot **KR16** del fabricante **KUKA**.

2.1 Definición de robot antropomórfico

Un robot antropomórfico es un robot industrial, descrita por la Asociación Internacional de Estándares (ISO) como: *Manipulador de 3 o más ejes, con control automático reprogramable, multiaplicación, móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial. Incluye al manipulador (sistema mecánico y accionadores) y al sistema de control (software y hardware de control y potencia).*

Por lo tanto, como indica “antropomórfico”, tiene aspecto físico parecido al ser humano, en el caso de los robots manipuladores, su parecido es en la parte del hombro, brazo y muñeca. Con esta morfología, podemos colocarnos en diferentes posiciones del espacio y orientar la herramienta para realizar un determinado trabajo. Esta configuración de robots industriales es usada para tareas pesadas y repetitivas.

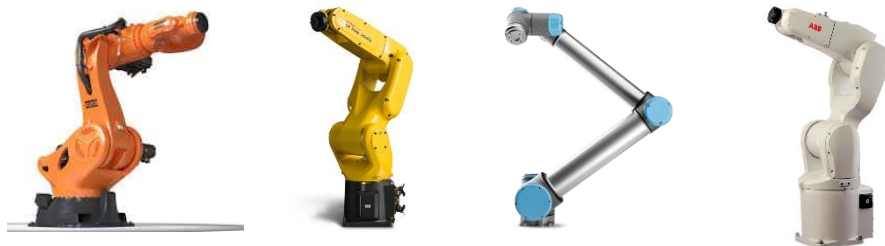


Figura 1: Robots antropomórficos (KUKA, FANUC, UNIVERSAL ROBOTS, ABB, respectivamente)

2.2 Características de un robot antropomórfico

En este apartado, se va a mencionar las características (prestaciones) de un robot antropomórfico, las cuales son:

-Grados de libertad: Es el movimiento en un espacio tridimensional, es decir, la capacidad de moverse hacia delante/atrás, arriba/abajo, izquierda/derecha (traslación en tres ejes perpendiculares), combinados con la rotación sobre tres ejes perpendiculares.

-Área de trabajo: Es el conjunto de puntos que pueden ser alcanzados por el punto central de la herramienta (TCP, "Tool Center Point), como podemos ver en la siguiente figura (figura 2), la cual nos muestra el área de trabajo del robot

-Capacidad de carga: Es el peso que puede transportar el manipulador.

-Velocidad: Se refiere a la velocidad lineal en los movimientos axiales del robot.

-Programabilidad: Es la capacidad para modificar la tarea mediante el cambio de programa, normalmente, los robots gozan de un elevado grado de autonomía, de modo que es capaz de hacer su tarea sin intervención del operador.

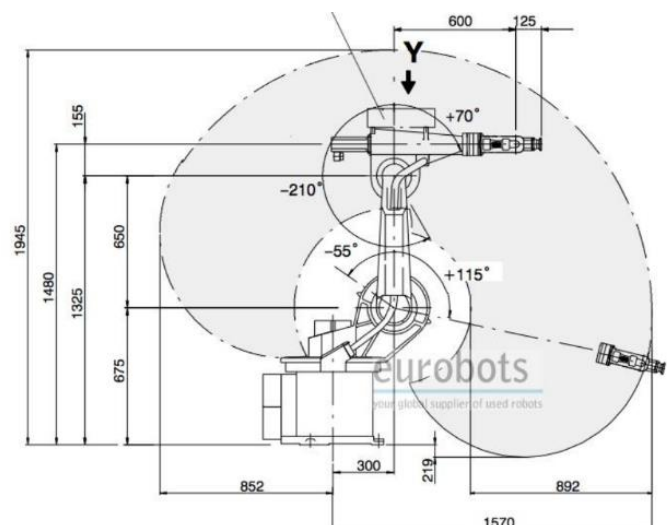


Figura 2: Área de trabajo del robot KUKA KR6

2.3 Morfología de un robot antropomórfico

La morfología de un robot antropomórfico está formada por varios eslabones, los cuales se unen mediante articulaciones con distintos grados de libertad, los componentes de un robot industrial según su función son los siguientes (Figura 3):

- **Brazo:** Realiza grandes movimientos.
- **Muñeca:** Adopta la orientación deseada.
- **Elemento terminal:** Es la mano, puede actuar como pinza o distintas herramientas.

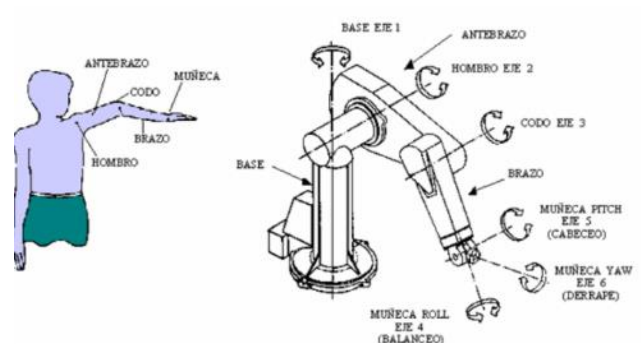


Figura 3: Similitud entre brazo y robot antropomórfico.

2.4 Elementos terminales de un robot

En este apartado, se mencionarán algunos elementos terminales, ubicados en el extremo final del mecanismo del robot, estos se encargan de realizar distintas tareas como: coger, pintar, soldar, pegar, etc.

Los elementos de sujeción (figura 4) más comunes son las llamadas “pinzas” (*Gripper* en inglés). Estas pueden ser de accionamiento eléctrico o neumático. El uso de diferentes modelos de pinzas se debe a distintos factores: si necesitamos una fuerza de agarre precisa, las pinzas de accionamiento neumático son las indicadas, si precisamos de una precisión en cierta distancia de agarre, las de accionamiento eléctrico son las indicadas.

Otro de los elementos de sujeción utilizado en la industria son las ventosas, las cuales usan el efecto Venturi o Coanda para crear el vacío entre la pinza y la ventosa, también, es común el uso de electroimanes.



Figura 4: Pinza por accionamiento eléctrico (izquierda) y elemento final de ventosa (derecha)

Los robots industriales también son utilizados para operaciones de soldadura, pintura o mecanizado. Para estos casos es necesario dotar al robot de la herramienta adecuada para estas operaciones, existen multitud de herramientas según la operación a realizar, en la siguiente tabla (Tabla 1) se muestran las herramientas dotadas en algunas operaciones.

Operación	Herramienta	Descripción gráfica
Paletizado	Pinzas, ventosas, etc.	
Pintura, barniz	Pistola de atomización	
Soldadura	Arco voltaico, TIG, MIG, electrodo, hilo, etc.	

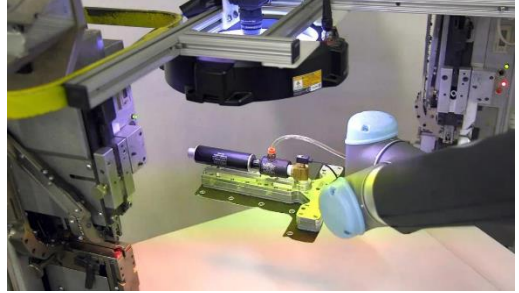
Mecanizado	Taladro, fresadora, sierra, etc.	
Control de calidad	Sensores, escáneres, Visión por computador, etc.	

Tabla 1: Algunas de las operaciones que pueden realizar los robots industriales.

3 DISEÑO DEL GEMELO DIGITAL (DT)

En este capítulo se mostrará el diseño del gemelo digital del *Franuc 24*, para ello, se ha utilizado *Solidworks* y *Matlab*. El gemelo digital (DT) es una realidad virtual del robot, gracias a este, se puede testear cualquier secuencia de movimientos (programa) antes de que lo ejecute el robot o mientras lo está ejecutando.

En este apartado, se va a mostrar el robot diseñado en formato “sólido”, este diseño se trata del primer diseño del robot, con el objetivo de crear una realidad virtual de un robot para así poder controlarlo a través de una interfaz gráfica de usuario. El robot consta de distintas partes: Base, brazo inferior, brazo superior, muñeca y pinza. (Figura 5).

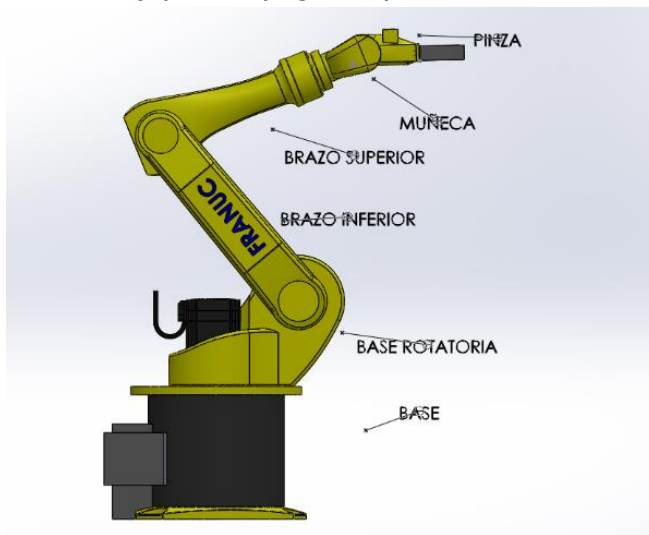


Figura 5: Partes del robot “FRANUC 24”.

Para la creación del mismo, se debe de seguir los siguientes pasos:

1: Diseñar el modelo CAD, para ello, se pueden utilizar distintos softwares como Solidworks, CATIA, Inventor, etc.

2: Crear “posición 0”, es importante colocar el robot en una posición inicial antes de crear el archivo .wrl, archivo necesario para la creación de la realidad virtual en Matlab.

3: Guardar el archivo en .wrl para importarlo en Matlab mediante el comando “smimport(robot.wrl)”, se puede ver en la siguiente figura (figura 6)

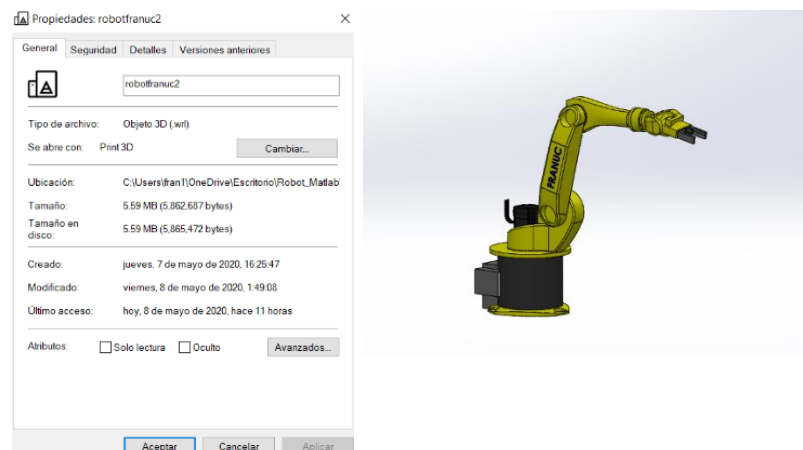


Figura 6: Creación del archivo .wrl

4: Se abre la herramienta “3D world editor” de Matlab y se crean los nodos del robot, agrupándolo en formato árbol. Una vez configurado lo dicho, se debe de definir la rotación y traslación de las distintas articulaciones, en la figura 7 se puede ver el antes/después tras realizar esta operación.

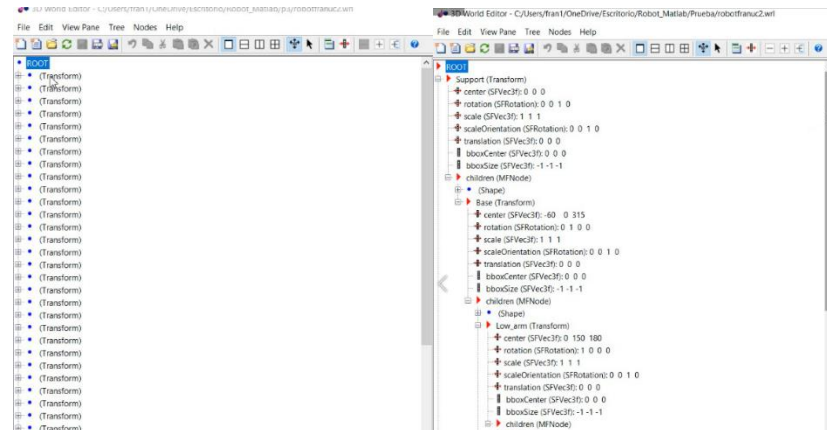


Figura 7: Antes vs después tras la configuración de la realidad virtual.

5: Una vez que ya está preparada la realidad virtual del robot, se pasa a programar la configuración de la realidad virtual para poder visualizarla, para ello, se empieza creando una “figura”, esta figura está compuesta por distintos nodos a los que se puede aplicar distintas rotaciones, en la figura 8 se puede ver como crear esta figura y configurar sus nodos.

```
function startupFcn(app)
    %Open figure
    app.myworld = vrworld('robotfranc2');
    open(app.myworld);
    app.ca= vr.canvas(app.myworld,gcf);
    set(app.myworld, 'Description', 'robotfranc2');

    %We define our nodes
    app.nodo1 = vrnode(app.myworld,'Base');
    app.nodo2 = vrnode(app.myworld,'Low_arm');
    app.nodo3 = vrnode(app.myworld,'High_arm');
    app.nodo4 = vrnode(app.myworld,'Wrist');
    app.nodo5 = vrnode(app.myworld,'Gripper');
    app.nodo6 = vrnode(app.myworld,'Gripper_1');
    app.nodo7 = vrnode(app.myworld,'Gripper_2');
```

Figura 8: configuración de la realidad virtual y creación de nodos

4. Cinemática del robot

En este capítulo se presentan los fundamentos teóricos utilizados para la realización de este trabajo, tarea indispensable para el correcto funcionamiento del *Franuc 24*, para ello, se empezará con una breve introducción sobre los conceptos necesarios para la realización de la fundamentación teórica, posteriormente, se calculará la cinemática directa del robot y la cinemática inversa, los límites en el área de trabajo del robot debido al diseño del mismo y, por último, la dinámica del robot, es decir, la relación de velocidades en las

distintas articulaciones para que el robot se mueva correctamente a través de un eje.

4.1 Conceptos básicos

En este apartado se explican los conocimientos necesarios para definir la cinemática de un robot, esto se consigue referenciando las velocidades y posiciones del robot respecto a un sistema de coordenadas. Se empezará explicando qué es un sistema de coordenadas y cómo se relaciona con las distintas posiciones del robot. Después se realizará un estudio de la formulación para conseguir referenciar los distintos sistemas de coordenadas. Posteriormente, se finaliza relacionando las velocidades del extremo con las distintas velocidades generadas en cada eje de rotación.

4.1.1 Sistema de coordenadas

Un sistema de coordenadas define un sistema coordenado bidimensional o tridimensional, partiendo de un punto fijo conocido como origen. Los objetos y las posiciones del robot se localizan mediante medidas a lo largo de los ejes de los sistemas de coordenadas, los robots utilizan distintos sistemas de coordenadas, cada uno tiene su punto de referencia:

-**Sistema de coordenadas mundo:** Tiene su punto cero en un punto fijo de la célula de trabajo.

-**Sistema de coordenadas de la base:** Este sistema tiene su punto cero en la base del robot, lo que resulta útil para calcular la posición en la que se encuentra el punto central de trabajo (TCP) del robot.

-**Sistema de coordenadas del objeto de trabajo:** Este sistema de referencia corresponde al plano de trabajo o pieza en el que el robot desempeña su función.

-TCP (Punto central de la herramienta): En inglés “Tool Center Point”, tiene su origen en el punto central de la herramienta seleccionada, definiendo la posición y la orientación de la misma, véase *figura 9*.

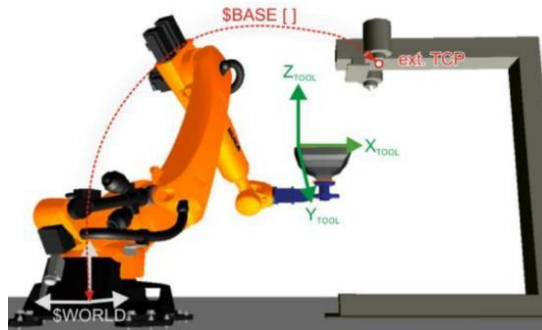


Figura 9: TCP de un robot

4.1.2 Matrices de transformación homogénea

Para conocer la posición y rotación de una cadena de sistemas de coordenadas, se va a utilizar el método de Denavit-Hartenberg, para la realización de este método, es necesario conocer qué son las matrices de transformación homogénea.

Se define matriz de transformación homogénea T a una matriz que expresa la posición y orientación de un sistema de coordenadas respecto a otro. Esta matriz está compuesta por 4 submatrices de distinto tamaño que representan la rotación, traslación, perspectiva y escalado.

$$T = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} \quad (\text{Ec 1})$$

$R_{3 \times 3}$: Matriz de rotación.

$p_{3 \times 1}$: Vector de traslación.

$f_{1 \times 3}$: Vector de perspectiva. Para robótica $f=(0,0,0)$.

$w_{1 \times 1}$: Factor de escala. Lo tomaremos como $w=1$.

4.2 Introducción al funcionamiento de Franuc 24

Antes de comenzar la resolución de la cinemática directa e inversa del robot, hay que saber los movimientos que se le demanda al robot. Primero se van a describir las articulaciones que forman el robot.

Se empieza con la articulación en la base, este eje se encargará de mover todo el brazo robótico de derecha a izquierda y viceversa, este eje modifica el **eje y** en el espacio de trabajo. Posteriormente, se encuentran las articulaciones en el “hombro y codo” del robot, que se encargan de moverse en el **eje x y z**. Por último, se encuentra la muñeca, formada por una articulación que permite girar la muñeca sobre sí misma y después se encuentra la articulación que permite girar la herramienta para posicionarla en un plano paralelo al suelo.

Por lo tanto, Franuc 24 está destinado para hacer movimientos en el **plano xz** y girarse sobre sí mismo, cambiando el ángulo del **plano xz**.

4.3 Cinemática directa del robot

El problema cinemático directo consiste en la representación de la posición y orientación del extremo final del robot conociendo las variables rotacionales de sus articulaciones. En el caso de *Franuc 24*, se tienen 5 ejes rotacionales, como se puede observar en la *figura 10*, el primer eje se encuentra en la base, se encarga de mover todo el robot en el plano xy, que corresponde al plano del suelo, el segundo se encuentra en el hombro del robot, es decir, en la base y se encargaría de mover todo el brazo en un plano xz, el tercero, corresponde al codo, se encarga de mover el brazo inferior (antebrazo), el cuarto se encuentra en la muñeca, pudiendo cambiar la orientación de la herramienta en el mismo plano que la base y, por último, se encuentra el eje de rotación de la pinza, que rotaría en un el eje xz.

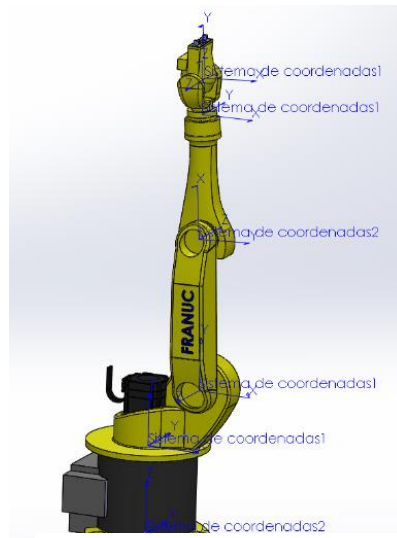


Figura 10: Robot “Franuc 24” y sus articulaciones

La relación que existe entre dos o varios elementos contiguos puede calcularse haciendo uso del sistema de referencia de cada uno de los elementos y calculando sus posiciones en cadena, cuando hay varios elementos se puede complicar este cálculo, por lo tanto, en robótica se utiliza el método de Denavit-Hartenberg, un método matricial que se propuso en 1955 por Denavit y Hartenberg, que establece la localización que debe tomar cada sistema de coordenadas ligado a cada eslabón de una cadena articulada, para poder sistematizar la obtención de las ecuaciones cinemáticas de la cadena completa.

Escogiendo los sistemas de coordenadas asociados a cada eslabón según la representación propuesta por D-H, será posible pasar de uno al siguiente mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.

Estas 4 transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento $i-1$ con el sistema del elemento i . Las transformaciones en cuestión son las siguientes:

1. Rotación alrededor del eje Z_{i-1} un ángulo θ_i .
2. Traslación a lo largo de Z_{i-1} una distancia d_i ; vector d_i $(0, 0, d_i)$.
3. Traslación a lo largo de X_i una distancia a_i ; vector a_i $(0, 0, a_i)$.
4. Rotación alrededor del eje X_i un ángulo α_i .

Dado que el producto de matrices no es conmutativo, las transformaciones se han de realizar en el orden indicado. De este modo se tiene que:

$${}^{i-1}\mathbf{A}_i = \text{Rot}_z(\theta_i) \cdot \text{T}(0,0,d_i) \cdot \text{T}(a_i,0,0) \cdot \text{Rot}_x(\alpha_i)$$

Realizando el producto de matrices obtenemos:

$$\begin{aligned} {}^{i-1}\mathbf{A}_i &= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} C\theta_i & -C\alpha \cdot S\theta_i & S\alpha \cdot S\theta_i & a_i \cdot C\theta_i \\ S\theta_i & C\alpha \cdot C\theta_i & -S\alpha \cdot C\theta_i & a_i \cdot S\theta_i \\ 0 & S\alpha & C\alpha & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Ec. 2}) \end{aligned}$$

Donde θ_i , d_i , a_i , α_i son los parámetros D-H del eslabón i . De este modo, basta con identificar los parámetros θ_i , d_i , a_i , α_i para obtener las matrices ${}^{i-1}\mathbf{A}_i$ y relacionar así todos y cada uno los eslabones del robot. Como se ha indicado, para que la ${}^{i-1}\mathbf{A}_i$ relacione los sistemas $\{\mathbf{S}_{i-1}\}$ y $\{\mathbf{S}_i\}$ es necesario que los sistemas se hayan escogido de acuerdo a unas determinadas normas. Estas normas junto con la definición de los cuatro parámetros de Denavit Hartenberg conforman el siguiente algoritmo para la resolución del problema cinemático directo que se comenta a continuación:

- 1) Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- 2) Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n .
- 3) Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.

- 4) Para i de 0 a $n-1$ situar el eje z_i sobre el eje de la articulación $i+1$.
- 5) Situar el origen del sistema de la base $\{S_0\}$ en cualquier punto del eje z_0 . Los ejes x_0 e y_0 se situarán de modo que formen un sistema dextrógiro con z_0 .
- 6) Para i de 1 a $n-1$, situar el sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fuesen paralelos $\{S_i\}$ se situaría en la articulación $i+1$.
- 7) Para i de 1 a $n-1$, situar x_i en la línea normal común a z_{i-1} y z_i .
- 8) Para i de 1 a $n-1$, situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .
- 9) Situar el sistema $\{S_n\}$ en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .
- 10) Obtener θ_i como el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.
- 11) Obtener d_i como la distancia, medida a lo largo de z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que x_i y x_{i-1} quedasen alineados.
- 12) Obtener a_i como la distancia medida a lo largo de x_i , que ahora coincidiría con x_{i-1} , que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$.
- 13) Obtener α_i como el ángulo que habría que girar en torno a x_i , que ahora coincidiría con x_{i-1} , para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$.
- 14) Obtener las matrices de transformación ${}^{i-1}A_i$.
- 15) Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{n-1}A_n$.
- 16) La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares.

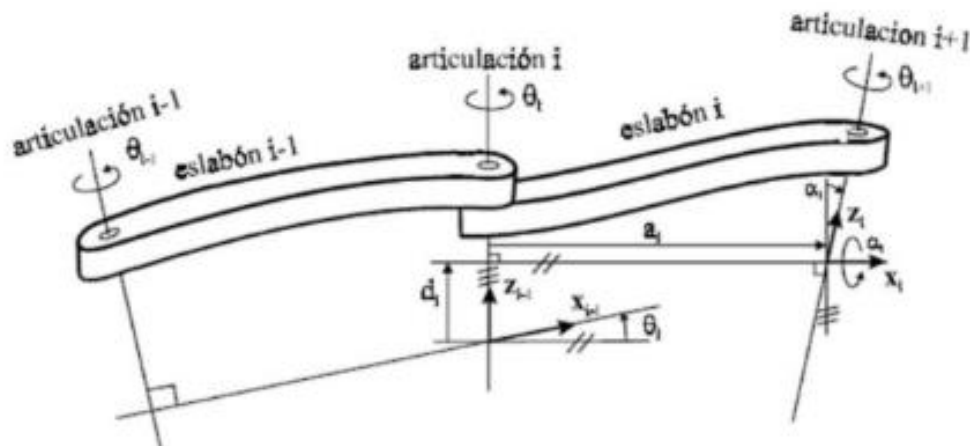


Figura 11: Parámetros Denavit-Hartenberg para un eslabón giratorio.

A continuación, se mostrará la tabla correspondiente a nuestros valores de Denavit-Hartenberg (Tabla 2), para la realización de esta tabla hay que tener en cuenta:

- θ_i : Es el ángulo que forman los ejes X_{i-1} y X_i medido en un plano perpendicular al eje Z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.
- d_i : Es la distancia a lo largo del eje Z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ ésimo hasta la intersección del eje Z_{i-1} con el eje X_i . Se trata de un parámetro variable en articulaciones prismáticas.
- a_i : Es la distancia a lo largo del eje X_i que va desde la intersección del eje Z_{i-1} con el eje X_i hasta el origen del sistema i ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes Z_{i-1} y Z_i .
- α_i : Es el ángulo de separación del eje Z_{i-1} y el eje Z_i , medido en un plano perpendicular al eje X_i , utilizando la regla de la mano derecha.

Link	θ_i	d_i	a_i	α_i
1	0	L0	0	0
2	Q_1	L1	L2	90
3	$Q_2 + 90$	0	L3	0
4	Q_3	0	L4	0
5	-90	0	0	-90
6	Q_4	L5	0	90
7	Q_5	0	0	-90

8	0	L6	0	0
---	---	----	---	---

Tabla 2: Parámetros Denavit-Hartenberg

Para realizar los parámetros del robot, ha sido necesario introducir ángulos de apoyo, en concreto se han hecho en los links 1,5 y 8, estos links son necesarios para la correcta orientación del robot, cuya estructura queda reflejada en la siguiente figura (figura 12)

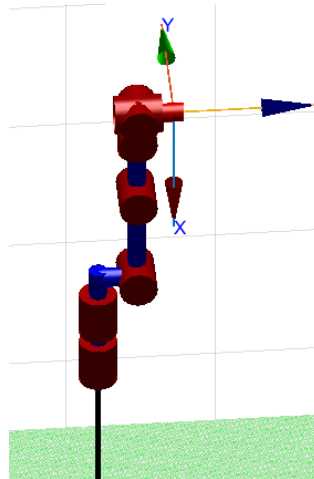


Figura 12: Estructura del robot Franuc 24.

Una vez que se tengan los parámetros D-H, se pasa a calcular la matriz de transformación T, que viene determinada por la siguiente ecuación (Ec.3):

$$T = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{n-1}A_n = {}^0A_n \text{ (Ec.3)}$$

Donde $T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o & a_x & p_x \\ n_z & o_x & a_x & p_x \\ 0 & 0 & 0 & w \end{bmatrix}$, pudiendo obtener los puntos en el espacio (p_x, p_y, p_z) .

4.4 Cinemática inversa del robot

El problema cinemático inverso consiste en encontrar los valores angulares que deben de tener las distintas articulaciones del robot para que su extremo se posicione y oriente según una determinada localización espacial. Este planteamiento es más difícil que el cinemático directo, ya que puede que no haya una solución para todos los puntos del espacio, para ello es necesario especificar las limitaciones en el espacio de nuestro robot

Teniendo la matriz de transformación homogénea T , es lógico pensar que podría manipularse la matriz para obtener la relación inversa, En la práctica, esta operación es compleja, sobretodo, para robots con determinados grados de libertad. En un robot de 6 ejes, manipular la matriz implica trabajar con 12 ecuaciones, aunque realmente sólo se tengan 6 incógnitas, por ello, se utilizan distintos métodos para la resolución del problema cinemático inverso.



Figura 13: Diagrama de relación entre cinemática directa e inversa.

El método utilizado para la resolución del problema cinemático del robot de este proyecto será el **método geométrico**, este método es muy apropiado para robots con pocos grados de libertad (hasta 3) debido a que hace uso de relaciones geométricas y trigonométricas entre los eslabones del robot. Sin embargo, esto resulta más complicado cuantos más grados de libertad tenga el robot, por ello, se ha dividido el robot en dos partes, el brazo y la muñeca, para facilitar la resolución con este método.

Para la resolución del problema cinemático inverso, se realizará la siguiente tabla (Tabla 3), la cual contendrá las incógnitas utilizadas en este proceso.

Tabla 3: Parámetros del robot

Parámetro	Descripción	Valor
Q1	Ángulo eje de la base	-
Q2	Ángulo eje del hombro	-
Q3	Ángulo eje del codo	-
Q4	Ángulo eje de la muñeca	-
Q5	Ángulo eje de la herramienta	-
L0	Altura de la base	196.5mm
L1	Altura desde la base al hombro en el eje z	147.5mm
L2	Distancia desde la base al hombro en el eje x	130mm
L3	Longitud desde el hombro al codo	340mm
L4	Longitud del antebrazo	268.58mm
L5	Longitud de la muñeca	75.24mm
L6	Longitud de la herramienta (TCP)	120.8mm

Px	Punto espacial en el eje X	-
Py	Punto espacial en el eje Y	-
Pz	Punto espacial en el eje Z	-

Los parámetros de entrada para la resolución del problema cinemático inverso son los puntos en el espacio (Px, Py, Pz), se empieza calculando el ángulo de la base (Q1), en el caso del Franuc 24, al conocer el ángulo de la base, no haría falta calcularlo, simplemente:

Q1 = ángulo de la base (el operario lo define).

Posteriormente, se va a calcular los ángulos de las articulaciones Q2 y Q3, para ello hay que tener en cuenta el teorema del coseno (figura 14) y aplicarlo a la estructura del robot (figura 15).

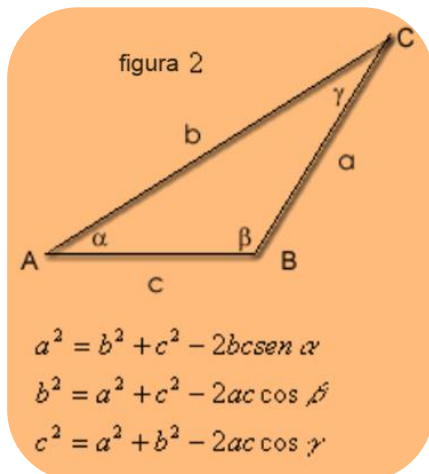


Figura 14: Teorema del coseno.

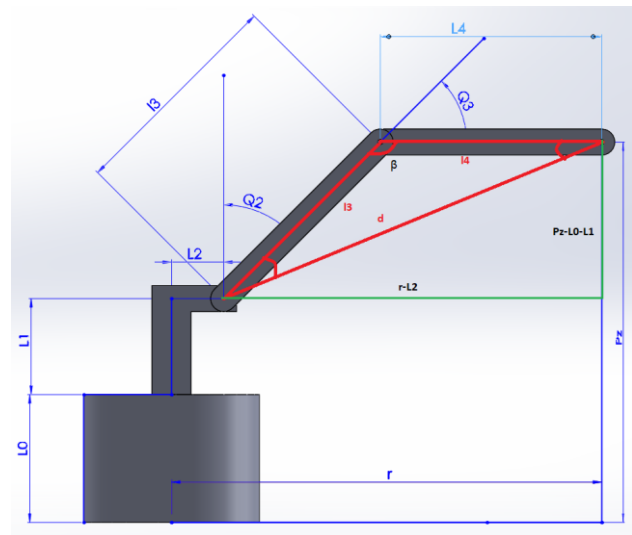


Figura 15: Estructura brazo del robot Franuc 24.

Lo primero que se va a calcular es la distancia "d", posteriormente se aplicará el teorema del coseno para obtener el ángulo "β" que permitirá obtener Q3:

$$r^2 = P_x^2 + P_y^2 \quad (\text{Ec. 4})$$

sustituyendo la ecuación 5.4 en la ecuación 5.5, se obtiene d^2

$$d^2 = (r-L_2)^2 + (P_z-L_0-L_1)^2 \quad (\text{Ec. 5})$$

Ahora se puede calcular β:

$$d^2 = (l_4 + l_5 + l_6)^2 + l_3^2 + 2 \cdot (l_4 + l_5 + l_6) \cdot l_3 \cdot \cos \beta \quad (\text{Ec. 6})$$

$$\cos \beta = \left(\frac{-l_3^2 - (l_4 + l_5 + l_6)^2 + d^2}{2 \cdot l_3 \cdot (l_4 + l_5 + l_6)} \right) \quad (\text{Ec. 7})$$

Por motivos de eficiencia a la hora de la computación y la programación del algoritmo, es mejor usar la expresión de la *arcotangente* en lugar del *arcoseno*, de este modo se obtiene:

$$\sin \beta = \pm \sqrt{1 - \cos^2 \beta} \quad (\text{Ec. 8})$$

$$\beta = \tan^{-1} \left(\frac{\pm \sqrt{1 - \cos^2 \beta}}{\cos \beta} \right), \text{ donde } \cos \beta = \left(\frac{-l_3^2 - l_4^2 + d^2}{2 \cdot l_3 \cdot l_4} \right) \quad (\text{Ec. 9})$$

Una vez que tenemos β , se calcula Q_3 :

$$Q_3 = 180 - \beta \quad (\text{Ec. 10})$$

Una vez se tiene Q_3 , se calcula Q_2 haciendo uso del teorema del seno (Figura 16) en la estructura del robot de la *figura 17*:

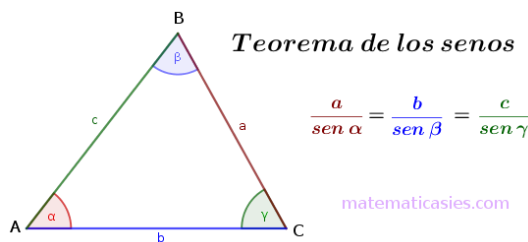


Figura 16: Teorema del seno.

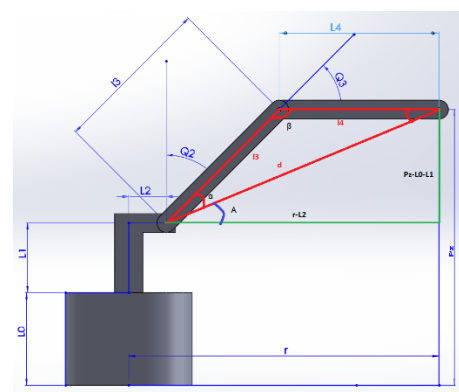


Figura 17: Estructura del robot.

Hay que sustituir las ecuaciones 13 y 14 en la ecuación 11, obteniendo de esta manera la ecuación final 15.

$$Q2 = 90 - \alpha - A \quad (\text{Ec. 11})$$

$$d^2 = (r - l_2)^2 + (P_z - l_0 - l_1)^2 \quad (\text{Ec. 12})$$

$$A = \tan^{-1} \left(\frac{(P_z - l_0 - l_1)^2}{(r - l_2)^2} \right) \quad (\text{Ec. 13})$$

$$\alpha = \tan^{-1} \left(\frac{l_4 \cdot \sin(Q3)}{l_3 + l_4 \cdot \cos(Q3)} \right) \quad (\text{Ec. 14})$$

$$Q2 = 90 - \tan^{-1} \left(\frac{l_4 \cdot \sin(Q3)}{l_3 + l_4 \cdot \cos(Q3)} \right) - \tan^{-1} \left(\frac{(P_z - l_0 - l_1)^2}{(P_x - l_2)^2} \right) \quad (\text{Ec. 15})$$

Una vez se tiene los ángulos de las articulaciones Q1, Q2 y Q3, se pasa a calcular el ángulo de la herramienta Q5, ya que se va a programar el robot para que el ataque de la herramienta al objeto sea con la herramienta en un plano paralelo al suelo, así como los movimientos del mismo, esto se calcula realizando la diferencia entre la posición horizontal de la herramienta y los ángulos de las articulaciones Q2 y Q3.

$$Q5: -90 - Q2 - Q3 \quad (\text{Ec.16})$$

4.5 Áreas de trabajo del robot

El robot tiene ciertas limitaciones a la hora de realizar movimientos, estas limitaciones se deben al diseño del robot, ya que si cualquiera articulación rota más grados de los estipulados en los límites, las articulaciones pueden chocar entre sí. Las limitaciones se han marcado haciendo uso de una simulación en *Solidworks*, con ayuda del diseño en 3D se ha podido realizar todo tipo de movimientos y ver qué límites definir.

Una vez obtenidos los límites angulares en cada articulación del robot, es necesario marcar los puntos críticos en el espacio y diferenciar entre dos áreas de trabajo, el área 1 el ángulo de la articulación Q2 será positivo respecto a la vertical y en el área 2 será negativo respecto a la vertical (figura 18 y figura 19).

Se define como punto crítico, los puntos intersección entre las líneas que definen los límites del robot.

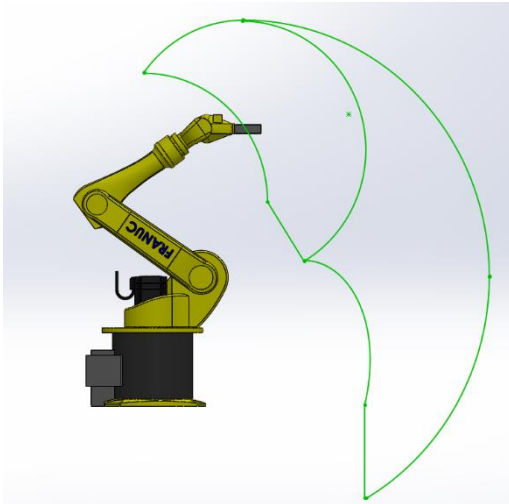


Figura 18: Áreas de trabajo en el plano XZ.

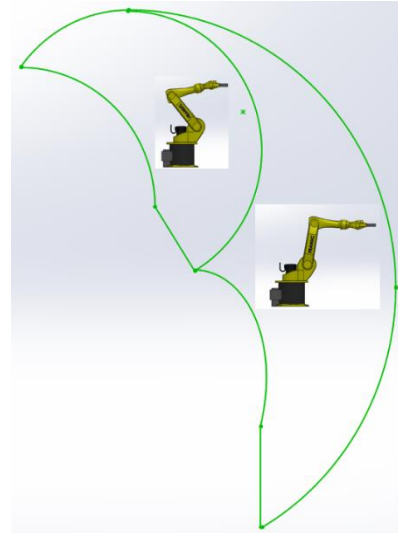


Figura 19: Posición del robot en las dos áreas.

Dentro de estas áreas, hay que saber que dentro de cada área hay puntos críticos, si el TCP sobrepasa la línea imaginaria de estos puntos críticos (tanto verticalmente como horizontalmente), la resolución de los límites cambia. Todas estas limitaciones tienen como objetivo que el robot trabaje (siempre que se pueda) con una formación en “codo arriba” (figura 20), que se respeten los límites de trabajo y por último, poder calcular la distancia restante hasta los límites máximos y mínimos tanto para el eje x como para el eje z. Para realizar todas estas operaciones, es preciso la ayuda de la tabla de variables para el cálculo de los límites (Tabla 4).

Nota importante: En los cálculos realizados teóricamente, todas las articulaciones tienen un desfase de 90° respecto a los cálculos realizados en Matlab, por ello, a lo largo de este capítulo nos encontraremos con los siguientes datos ($90 + \text{ángulo}$).

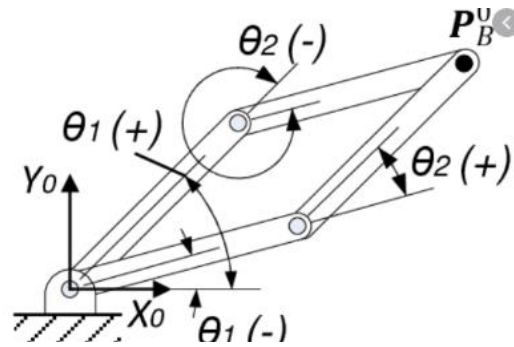


Figura 20: Soluciones codo arriba y codo abajo.

Tabla 4: Variables para el cálculo de los límites

Variable	Descripción	Valor en Franuc 24
Amin3	Ángulo mínimo articulación 3	-60
Amax2	Ángulo máximo articulación 2	90+54
Amin2	Ángulo mínimo articulación 2	-60
C1	Condición 1: Línea fronteriza entre áreas eje x	-
C2	Condición 2: Límite fronterizo entre áreas eje z	-
C3	Punto x en intersección superior entre área 1 y 2	-
C4	Punto z superior en intersección entre área 1 y 2	-
C5	Px límite izquierdo en área 2	-
C6	Pz límite izquierdo en área 2	-
C7	Punto z inferior en intersección entre áreas	-
C8	Punto x inferior en línea intersección entre áreas	-
C9	Punto x final inferior área 2	-
C10	Punto z final inferior área 2	-
Ar1	Condición de área 1	
Ar2	Condición de área 2	
QAZ	angulo entre la vertical y la recta límite inferior trasera del área 2	-

Primero se va a definir la línea fronteriza entre las dos áreas de trabajo en el plano XZ:

Límite entre áreas:

- C1: $\sqrt{(Px - L2 - L6)^2 + (Pz - L0 - L1 - L3)^2} = L4 + L5 \rightarrow$ Línea fronteriza entre áreas
- C2: $L0 + L1 + L3 + (L4 + L5) * \sin(A_{min3}) \leq Pz \leq L0 + L1 + L3 + L4 + L5 \rightarrow$ Condición en altura para la línea fronteriza

Para calcular si un punto pertenece al área 1 o al área 2, se va a calcular el punto **P_{xl}** para una altura **z** en la línea fronteriza con ayuda de la condición C1 y así comparar el punto **P_x** con el punto **x** obtenido:

$$\sqrt{(Px_l - L2 - L6)^2 + (Pz - L0 - L1 - L3)^2} = L4 + L5 \quad (\text{Ec. 17})$$

$$(Px_l - L2 - L6)^2 + (Pz - L0 - L1 - L3)^2 = (L4 + L5)^2 \quad (\text{Ec. 18})$$

$$(Px_l - L2 - L6)^2 = (L4 + L5)^2 - (Pz - L0 - L1 - L3)^2 \quad (\text{Ec. 19})$$

$$(Px_l - L2 - L6) = \sqrt{(L4 + L5)^2 - (Pz - L0 - L1 - L3)^2} \quad (\text{Ec. 20})$$

$$Px_l = \sqrt{(L4 + L5)^2 - (Pz - L0 - L1 - L3)^2} + L2 + L6 \quad (\text{Ec. 21})$$

Una vez se tiene **x**, si **P_x ≥ P_{xl}**, el TCP se encuentra en el área 2 (**Ar₂ = 1**), de lo contrario (**P_x < P_{xl}**), el TCP se encuentra en el área 1 (**Ar₁ = 1**).

Una vez separadas las áreas de trabajo, se van a definir los distintos puntos y restricciones en cada una de las áreas:

➤ **Condiciones en el área 1:**

- Ar₁: 1 si estamos dentro de área 1
- C3: $L2 + L6 \rightarrow$ Punto x en intersección superior entre área 1 y 2
- C4: $L0 + L1 + L3 + L4 + L5 \rightarrow$ Punto z intersección superior entre área 1 y 2
- C5: $L2 + L3 * \cos(A_{max2}) + L6 \rightarrow$ P_x límite izquierdo en área 2
- C6: $L0 + L1 + L3 * \sin(A_{max2}) + L4 + L5 \rightarrow$ P_z límite izquierdo en área 2

➤ **Condiciones en el área 2:**

- Ar2: 1 si se encuentra dentro de área 2
- C7: $L0 + L1 + L3 + (L4 + L5) \cdot \sin(A_{min3}) \rightarrow$ Punto z inferior en intersección entre áreas
- C8: $L2 + (L4 + L5) \cdot \cos(A_{min3}) + L6 \rightarrow$ Punto x inferior en línea intersección entre áreas
- C9: $L2 + (L3 + L4 + L5) \cdot \cos(A_{min2}) \rightarrow$ Punto x final inferior área 2
- C10: $L0 + L1 + (L3 + L4 + L5) \cdot \sin(A_{min2}) \rightarrow$ Punto z final inferior área 2
- QAZ: $\tan^{-1} \left(\frac{C9-C8}{C10-C7} \right) \rightarrow$ ángulo entre la vertical y la recta límite inferior trasera del área 2.

Cálculo de máximos y mínimos:

Una vez definidos todos los límites y las restricciones en las áreas, se pasa a calcular Pzmax, Pzmin, Pxmax y Pxmin.

Nota: Para calcular los puntos máximos, se va a despejar los ángulos máximos permitidos para cierto punto x o z en las funciones límites de área.

➤ **Pzmax:**

Para calcular los máximos en z, hay que tener en cuenta el área en la que el TCP se encuentra.

Primero, en el área 1, hay dos funciones límite para calcular Pzmax, las cuales hay que diferenciar a partir de un punto x, ese punto es "C3" (**mirar condiciones en Área 1**) .

- Si (**Ar1 & Px >= C3**) || Ar2

$$Px - L2 - L6 = (L3 + L4 + L5) \cdot \cos(QPzmax) \quad (\text{Ec. 22})$$

$$Pzmax - L0 - L1 = (L3 + L4 + L5) \cdot \sin(QPzmax) \quad (\text{Ec 23})$$

$$Pzmax = (L3 + L4 + L5) \cdot \sin(QPzmax) + L0 + L1 \quad (\text{Ec 24})$$

- Si $Ar1 \& Px < C3$, en esta zona del área 1, la función que define el límite en z es:

$$x = L2 + L3 \cdot \cos(A_{max2} - t) + L6$$

$$z = L0 + L1 + L3 \cdot \sin(A_{max2} - t) + L4 + L5$$

Donde $t: \{0 \dots 54\}$

Para calcular Pz_{max} , vamos a calcular el ángulo del punto máximo en x y utilizarlo para sacar el punto máximo en z:

$$Px = L2 + L3 \cdot \cos(QPz_{max}) + L6 \quad (\text{Ec. 25})$$

$$Pz_{max} = L0 + L1 + L3 \cdot \sin(QPz_{max}) + L4 + L5 \quad (\text{Ec. 26})$$

➤ **Px_{max}:**

Para calcular los máximos en x no es necesario conocer en qué área estamos, ya que sólo depende de una función límite máximo:

$$Pz - L0 - L1 = (L3 + L4 + L5) \cdot \sin(QPx_{max}) \quad (\text{Ec. 27})$$

$$Px_{max} - L2 - L6 = (L3 + L4 + L5) \cdot \cos(QPx_{max}) \quad (\text{Ec. 28})$$

$$Px_{max} = L2 + L6 + (L3 + L4 + L5) \cdot \cos(QPx_{max}) \quad (\text{Ec. 29})$$

➤ **Px_{min}:**

Para calcular el Px_{min} , hay que tener en cuenta el área y la zona en la que se encuentra el TCP, primero se va a empezar por el área 1, en la que se encuentran dos zonas, que se pueden separar a partir el de punto z "C6",

este punto también coincide con la separación de una de las tres zonas del área 2 para el cálculo de P_{xmin} , al igual que pasa con el punto "C2":

- Si $P_z \geq C6$

$$P_z = L0 + L1 + L3 \cdot \sin(QP_{xmin}) + L4 + L5 \quad (\text{Ec. 30})$$

$$P_{xmin} = L2 + L3 \cdot \cos(QP_{xmin}) + L6 \quad (\text{Ec. 31})$$

- Si $P_z < C6$ || $P_z \geq C7$

El límite en esta zona viene dado por la ecuación $z=f(x) = m \cdot x + n$,

donde $m = \frac{z2-z1}{x2-x1}$

$$z_1 = L0 + L1 + L3 \cdot \sin(A_{max2}) + L4 + L5 \quad (\text{Ec. 32})$$

$$z_2 = L0 + L1 + L3 \cdot \sin(A_{min3}) + L4 + L5 \quad (\text{Ec. 33})$$

$$x_1 = L2 + L3 \cdot \cos(A_{max2}) + L6 \quad (\text{Ec. 34})$$

$$x_2 = L2 + L3 \cdot \cos(A_{min3}) + L6 \quad (\text{Ec. 35})$$

Para calcular "n", se debe de ver qué ángulo tendrá L3 cuando $x = 0$:

$$0 = L2 + L3 \cdot \cos(\theta) + L6 \quad (\text{Ec. 36})$$

$$n = L0 + L1 + L3 \cdot \sin(\theta) + L4 + L5 \quad (\text{Ec. 37})$$

Por lo tanto, P_{xmin} viene dado por: $P_{xmin} = (z-n)/m$ (Ec. 38)

- Si $P_z < C7$

Para calcular el P_{xmin} en esta zona del área 2, hay que hacer uso de trigonometría, el punto 1 va a ser el punto inferior de la intersección entre ambas áreas:

$$(P_z - C7) = d_i \cdot \cos(QA_z) \rightarrow \text{obtenemos "d_i"} \quad (\text{Ec. 39})$$

$$Px_{min} = C8 + d_i \sin(QAz)$$

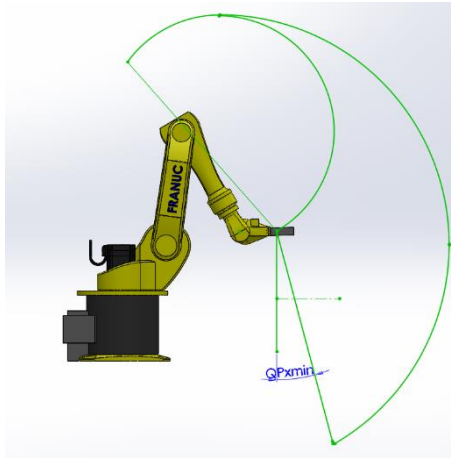


Figura 21: Ángulo QPxmin

➤ **Pzmin:**

Para calcular el Pzmin hay que tener en cuenta la zona en la que se encuentra el TCP, la cual viene separada por el punto C8, que es el punto Px inferior de la línea de limitación entre áreas:

- Si **Px >= C9**

$$Px - L2 - L6 = (L3 + L4 + L5) \cos(-QPzmin) \quad (\text{Ec. 41})$$

$$Pzmin - L0 - L1 = (L3 + L4 + L5) \sin(-QPzmin) \quad (\text{Ec. 42})$$

$$Pzmin = (L3 + L4 + L5) \sin(QPzmax) + L0 + L1 \quad (\text{Ec. 43})$$

- Si **C8 < Px < C9**

$$(Px - C8) = d \sin(QAz) \rightarrow \text{obtenemos "d"} \quad (\text{Ec. 44})$$

$$Pzmin = C7 + d \cos(QAz) \quad (\text{Ec. 45})$$

- Si **Px <= C8**

El límite en esta zona viene dado por la ecuación $z=f(x) = m \cdot x + n$,
donde

$$m = \frac{z_2 - z_1}{x_2 - x_1}$$

$$z_1 = L_0 + L_1 + L_3 \sin(A_{\max 2}) + L_4 + L_5 \quad (\text{Ec. 46})$$

$$z_2 = L_0 + L_1 + L_3 \sin(A_{\min 3}) + L_4 + L_5 \quad (\text{Ec. 47})$$

$$x_1 = L_2 + L_3 \cos(A_{\max 2}) + L_6 \quad (\text{Ec. 48})$$

$$x_2 = L_2 + L_3 \cos(A_{\min 3}) + L_6 \quad (\text{Ec. 49})$$

Para calcular “n”, se debe de ver qué ángulo tendrá L3 cuando $x = 0$:

$$0 = L_2 + L_3 \cos(\theta) + L_6 \quad (\text{Ec. 50})$$

$$n = L_0 + L_1 + L_3 \sin(\theta) + L_4 + L_5 \quad (\text{Ec. 51})$$

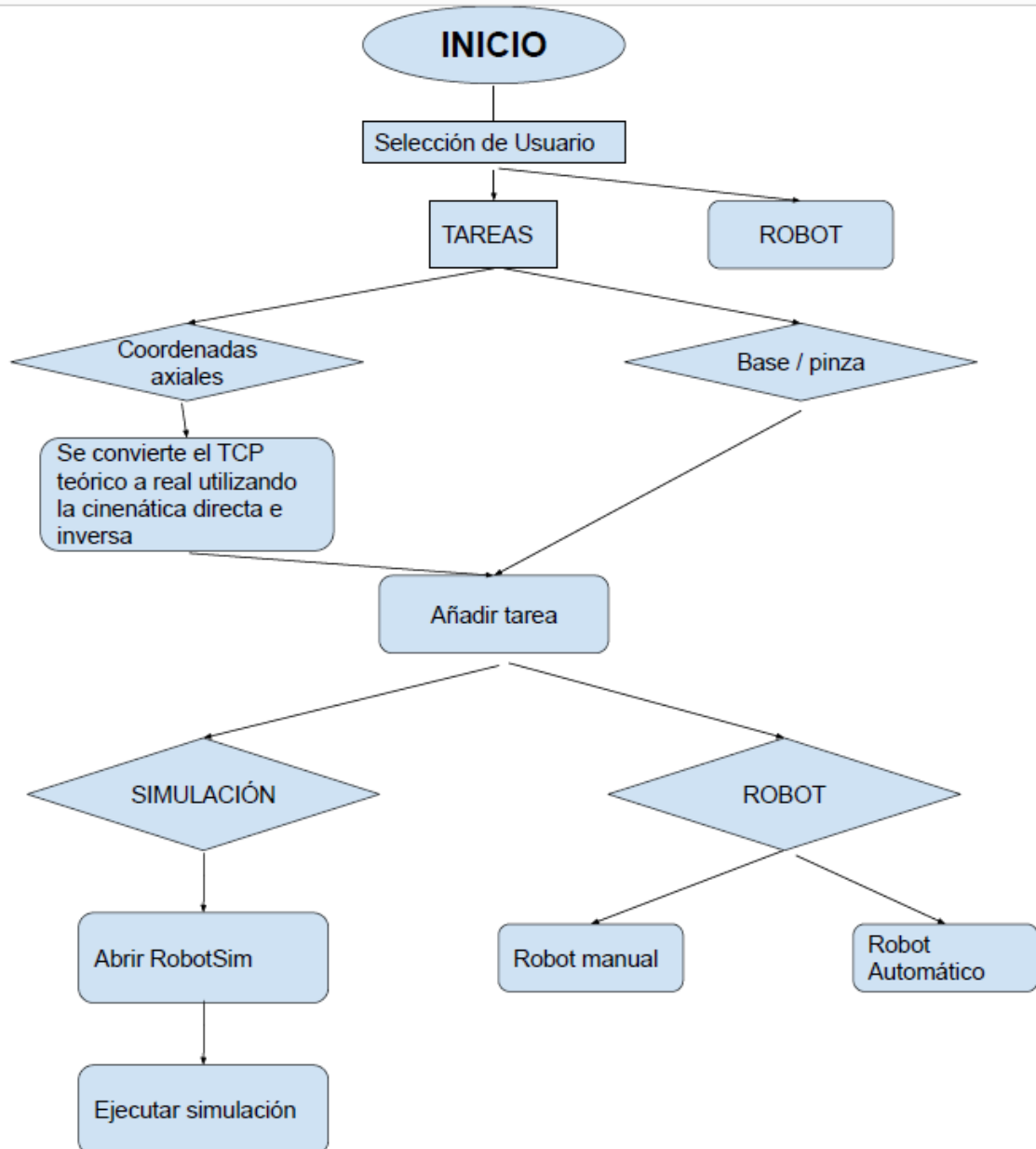
$$P_{x\min} \text{ viene dado por: } P_{z\min} = m \cdot P_x + n \quad (\text{Ec. 52})$$

5. ENTORNO DE DESARROLLO INTEGRADO PARA FRANUC 24

En este capítulo se presenta el entorno de desarrollo integrado para el robot, se trata de una interfaz gráfica de usuario creada a través de **AppDesigner** de Matlab, la GUI consta de 4 pantallas seleccionables por el usuario, las cuales son: INICIO, TAREAS, SIMULACIÓN y ROBOT.

5.1 Fases de la programación

En este apartado, se muestra un flujograma en el cual se irán explicando las distintas fases del programa:



- **INICIO:** La interfaz gráfica de usuario empieza en la pantalla de INICIO.
- **Selección de usuario:** Se selecciona el usuario y se proporciona el acceso a las aplicaciones de Tareas y Robot.
- **Robot:** En este primer acceso a robot sin haber pasado por Tareas, el robot sólo podrá funcionar manualmente, ya que no hay ningún programa guardado.

- **Tareas:** Una vez dentro de tareas, debemos de elegir movimientos de un tipo, ya pueden ser axiales, girar base o abrir/cerrar pinza.
- **Base/pinza:** Se selecciona los movimientos deseados, ya sea en la base o en la pinza, pero sólo puede tomarse la decisión en una de las dos opciones.
- **Coordenadas axiales:** Se introducen las coordenadas en los ejes x o z.
- **TCP:** En este programa, se calcula el TCP teórico según las coordenadas introducidas, mediante cinemática inversa se calcula la posición de los motores, se aproxima a un múltiplo de 1.8° (el paso de los motores) y se vuelve a calcular mediante cinemática inversa el TCP real.
- **Añadir tarea:** Se añaden las tareas a realizar en el programa de tareas. Una vez se tiene la tarea programada, se puede acceder a la aplicación de Simulación o Robot.
- **Simulación:** Aplicación encargada de la gestión de la simulación
- **AbrirRobotSim:** Se abre el gemelo digital del robot.
- **Ejecutar simulación:** Ejecuta la simulación de las tareas programadas.
- **Robot:** Aplicación encargada de la gestión del robot físico.
- **Robot manual:** Se puede mover el robot manualmente sin programación previa.
- **Robot automático:** Programa que ejecuta las tareas programadas en el robot físico.

5.2 Pantalla de INICIO

En esta primera pantalla, se encuentra la introducción al controlador del robot FRANUC 24, como muestra la siguiente figura (figura 7.1), se encuentran los distintos usuarios registrados con una validación a través de una contraseña. Una vez validado, la luz verde en el lateral superior derecho indica que se puede acceder a las demás pantallas.



Figura 22: Pantalla de inicio (con usuario aún sin acceder).

5.3 Pantalla de TAREAS

En la segunda pantalla, se encuentra la pantalla encargada de la selección de programa o programación de tareas (figura 7.2), encargada, como se puede deducir por el título, de la gestión de tareas del robot.



Figura 23: Pantalla de tareas.

En esta pantalla se puede programar las tareas a realizar por el robot para así, poder simularlas o ejecutarlas, para una correcta programación de tareas es imprescindible leer el **MANUAL DE USUARIO**, para un correcto funcionamiento de la simulación o del robot.

En la pantalla de tareas encontramos los siguientes cuadros de control y marcadores:

- **INICIO, SIMULACIÓN y ROBOT:** Estos pulsadores verdes se encargan de abrir las siguientes pantallas del programa, mostrando en naranja, la pantalla en la que el usuario se encuentra.
- **Cuadro de control de coordenadas axiales:** En este cuadro, se encuentran las instrucciones axiales, tanto en el **eje X** como en el **eje Z**, encima y debajo de los botones de control de los ejes, se encuentran los marcadores de los **límites superior e inferior** de los respectivos ejes. Después se puede ver los marcadores de **Pxreal** y **Pzreal**, en los que, mediante aproximación posible de la rotación de las distintas articulaciones, se ha conseguido calcular la posición real en los ejes x y z con el método de Denavit-Hartenberg. Finalmente, este cuadro consta del **error relativo** de la posición y los pulsadores de convertir y añadir movimiento.
- **Cuadro de la pinza:** En este cuadro se encuentran los botones de abrir y cerrar pinza, posteriormente hay que añadir el movimiento pulsando “añadir” en el cuadro de control.
- **Comandos Base y muñeca:** Estos comandos son los responsables de la creación de la tarea de mover la base y la muñeca.

- **Cuadro de información de articulaciones:** En este cuadro se pueden ver los valores de las articulaciones en grados, tanto teóricos como reales.

- **Tabla de programa:** En esta tabla se encuentran las tareas a realizar por la simulación o por el robot, acompañado de los botones “reset” y “eliminar movimiento” el cual elimina el último movimiento registrado.

Para poder pasar a la pantalla de SIMULACIÓN, es imprescindible tener un programa de tareas registrado.

5.4 Pantalla de SIMULACIÓN

En la tercera pantalla, se encuentra la pantalla simulación (figura 7.3.1), en la que se puede visualizar el programa de tareas que se puede simular a través de una tabla y los botones de “ABRIR ROBOTsim” el cual abre el gemelo digital del robot Franuc 24 (figura 7.3.2) y el botón de “SIMULAR”, el cual ejecuta la simulación.



Figura 24: Pantalla de SIMULACIÓN

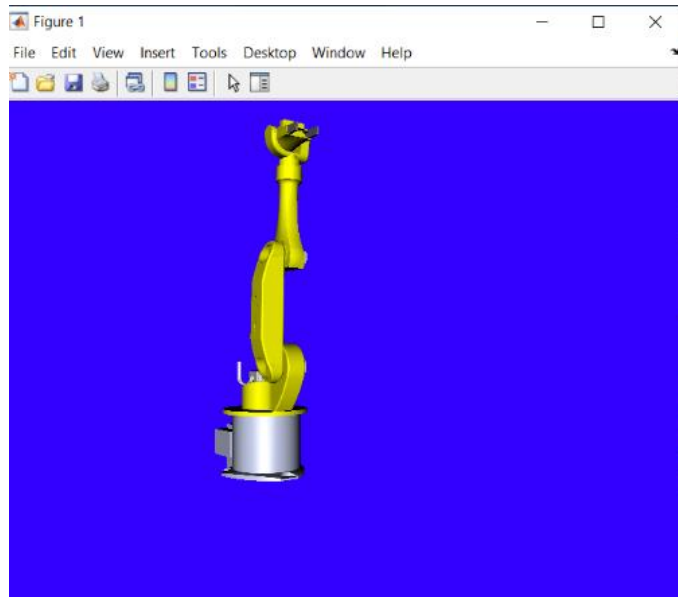


Figura 25: ROBOTsim

5.5 Pantalla de ROBOT

Por último, se encuentra la pantalla de robot (figura 7.4), en la que se pueden diferenciar 3 partes, la primera parte, el cuadro de comunicación, en el que establecemos la comunicación con algunos de los controladores existentes, en segundo lugar, se encuentran los comandos para controlar manualmente las distintas articulaciones. Para finalizar, se dispone de una tabla en la que se puede visualizar el programa que se puede ejecutar en el robot junto a los botones de posición inicial y de ejecución de programa.

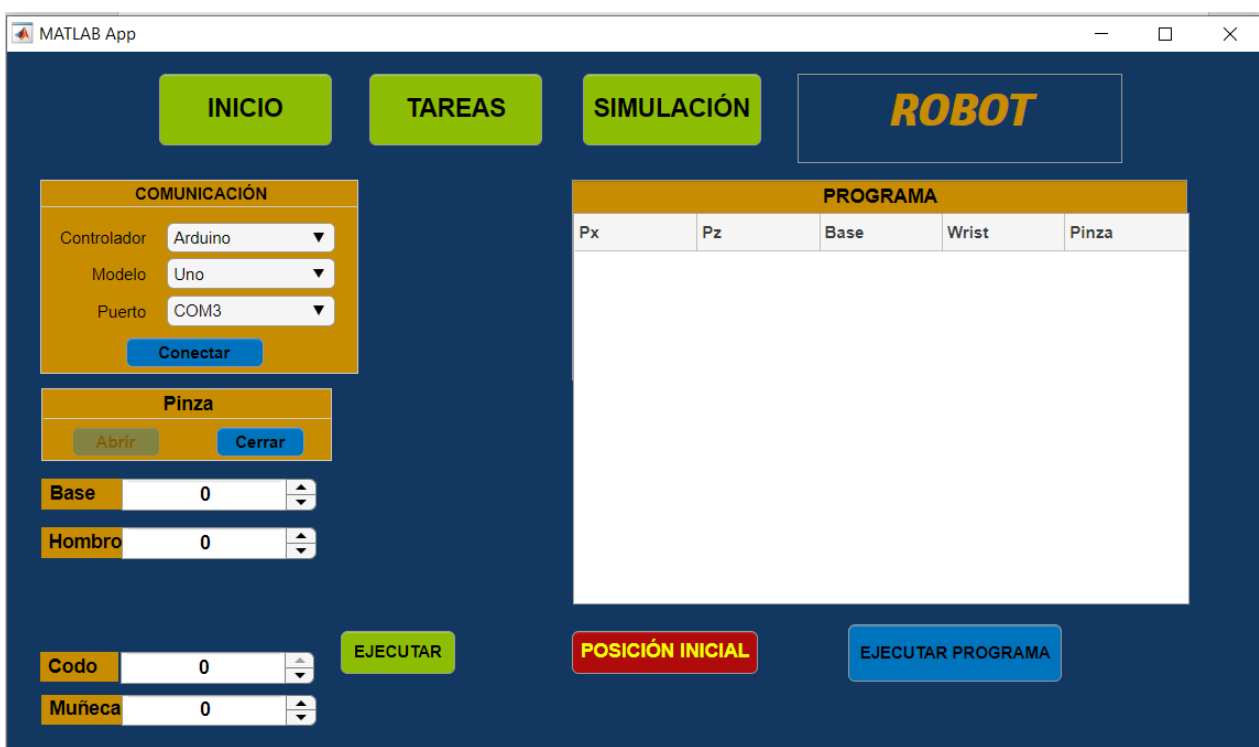


Figura 26: Pantalla de ROBOT.

6. ROBOT “FRANUC 24”

En este capítulo, se realiza la puesta en marcha del robot, para ello se ha diseñado la versión imprimible del robot, ya que para la simulación se ha utilizado una versión totalmente sólida, también, se muestra el diseño del control de potencia del robot y la configuración del protocolo de comunicación con el controlador.

6.1 Diseño imprimible del robot.

En este apartado se va a presentar la versión imprimible del robot, se ha tenido que realizar un escalado de un 60% del robot, consiguiendo un menor peso y una menor distancia de las articulaciones, con objetivo de reducir los momentos de fuerza que serán ejercidos sobre los ejes de los motores. También se ha conseguido reducir notablemente el tiempo de impresión, se ha utilizado una impresora “Ender 3 Pro”, utilizando PLA de 1.75 mm de diámetro, configurando la impresora con un relleno del 10%, velocidad de 40mm/s y altura de la capa de 0.2mm.

Los planos del robot se encuentran disponibles en el ANEXO I: Diseño robot “FRANUC 24”.

6.1.1 Base

La base del robot consta de 2 partes, la base fija y la base rotatoria, la base rotatoria se une a un cilindro móvil que se encuentra en la base fija, cuya función es la misma que la de un rodamiento. El motor paso a paso Nema17 se encuentra fijado en la base fija y moverá el cilindro móvil gracias a un sistema de engranajes interior-exterior como se muestra en la siguiente figura (figura 6.1.1)

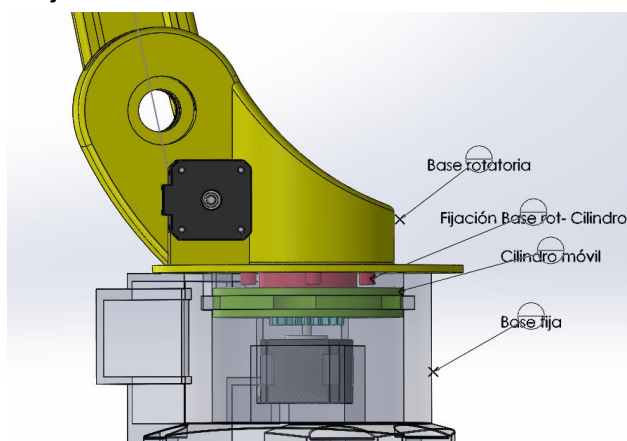


Figura 27: Mecanismo base.

6.1.2 Brazo inferior

El brazo inferior realizará el movimiento equivalente al que realiza un hombro, este irá sujeto a través de un eje situado en la base rotatoria, tal y como muestra la *figura 6.1.1*. En esta articulación, se encuentra implementado el sistema de polea-correa dentada encargado de mover la articulación del brazo superior (codo). (Figura 6.1.2)

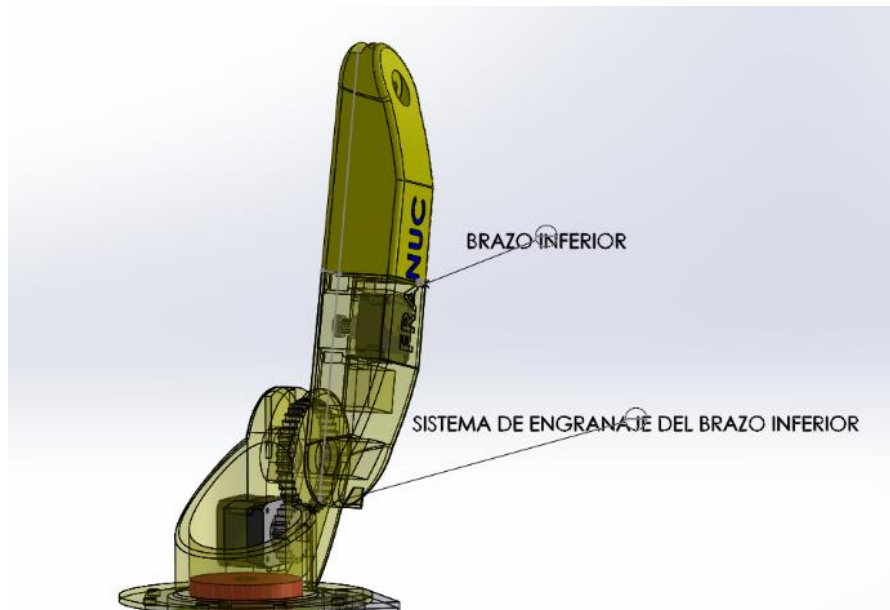


Figura 28: Mecanismo brazo inferior.

6.1.3 Brazo superior

El brazo superior realizará el movimiento equivalente al que realiza un codo, este irá sujeto a través de un eje situado en el brazo inferior, este eje será movido gracias a un sistema de correa-polea.

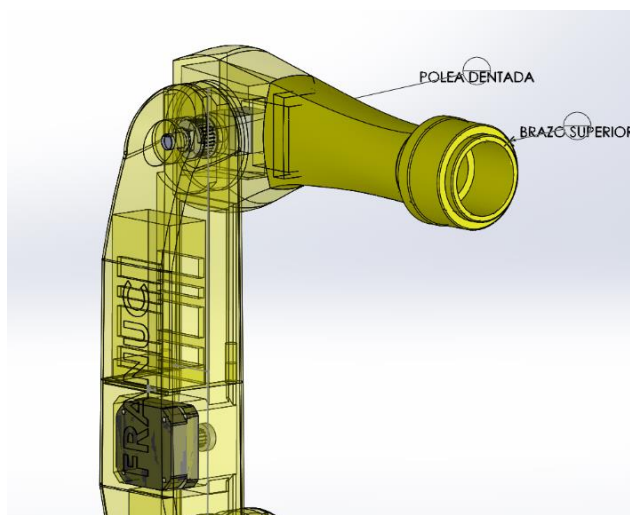


Figura 29: Mecanismo brazo superior.

6.1.4 Muñeca y pinza

Por último, se encuentra la muñeca y la pinza, la muñeca se encargará de proporcionar un eje a la pinza (figura 6.1.4), en la muñeca, se encuentra un motor, encargado de girar la pinza. En la pinza se encuentra un mecanismo que se encarga de abrir/cerrar la pinza, formado por unos brazos articulados que se encargan de transformar el movimiento rotatorio de un servomotor en un movimiento rectilíneo (figura 6.1.5)

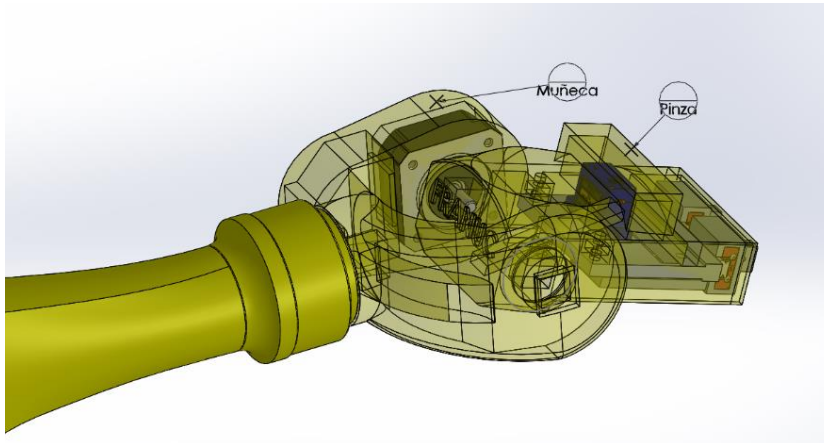


Figura 30: Mecanismo situado en la muñeca.

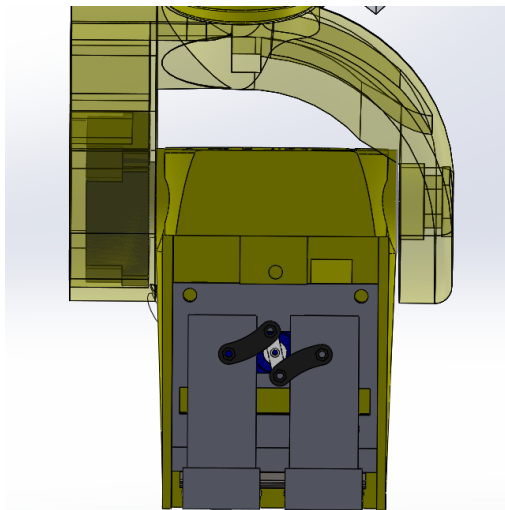


Figura 31: Mecanismo de la pinza.

6.2. Componentes hardware

En este apartado se muestra el control de potencia del robot, para el diseño del control de potencia se ha utilizado un Arduino UNO, 4 motores paso a paso Nema 17, un servomotor SG90, 4 drivers drv8825 para el correcto control de los motores paso a paso, una fuente de alimentación de 24V y 4 condensadores de 100uF. Los datasheets de los distintos componentes se pueden ver en el Anexo II: Datasheets.

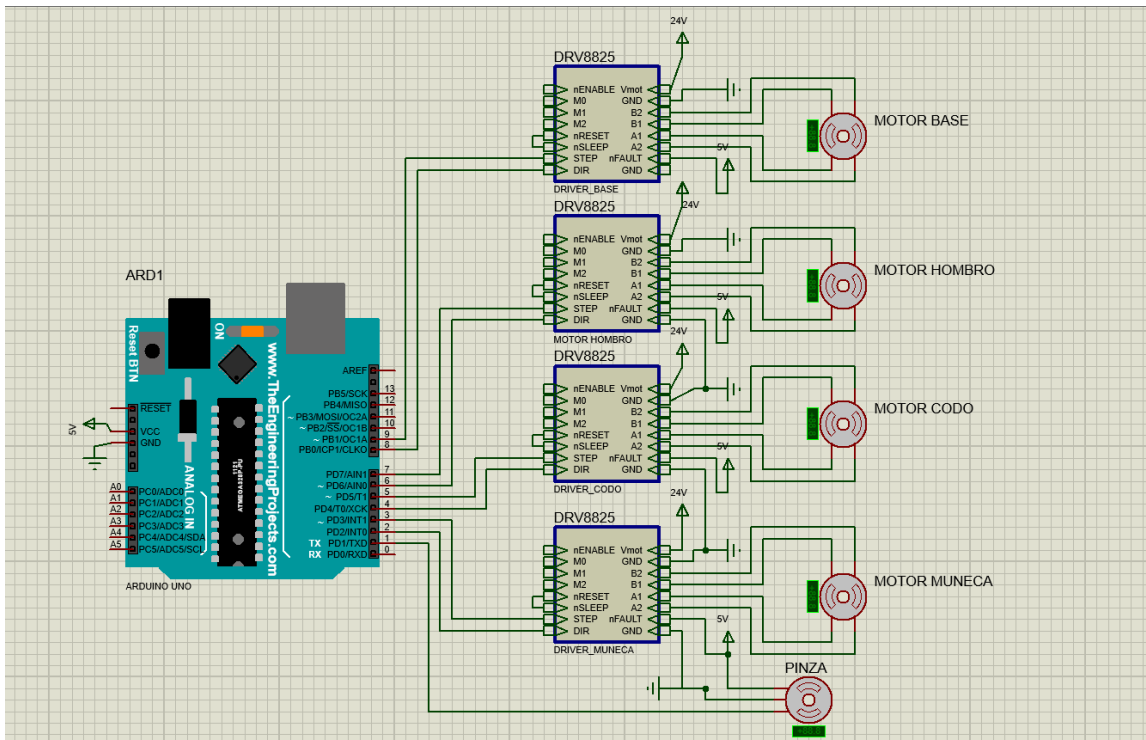


Figura 32: Control de potencia del robot “Franuc 24”.

El conexionado se ha realizado siguiendo las indicaciones del fabricante, en el caso del driver drv8825 se ha seguido el siguiente esquema:

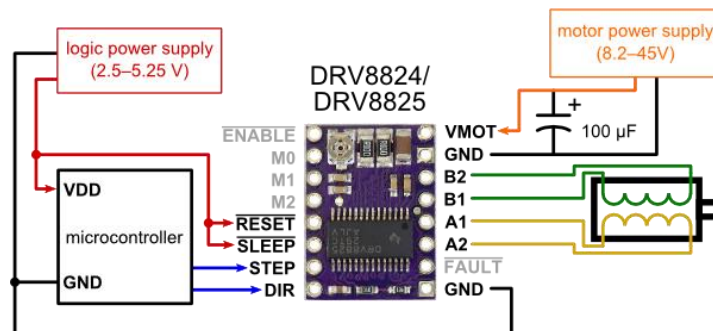


Figura 33: Conexionado DRV8825

LISTADO DE COMPONENTES:**Arduino UNO**

El Arduino Uno es una placa de microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc.¹² La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos. La placa tiene 14 pines digitales, 6 pines analógicos y programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB tipo B.



Figura 34: Arduino UNO R3

Driver DRV8825

Se trata de un driver para motores paso a paso, estos controladores permiten manejar los voltajes e intensidades a los que se está suministrando al motor para así controlar la velocidad de giro. Además, sirven como método de protección para evitar que la electrónica de los motores pueda resultar dañada limitando la corriente que circula (chopping).

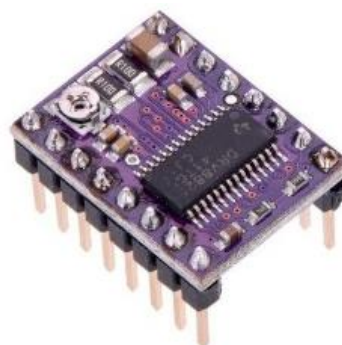


Figura 35: Driver DRV8825

Fuente alimentación 24V

La de alimentación (PSU en inglés) es el dispositivo que se encarga de transformar la corriente alterna de la línea eléctrica comercial que se recibe en los domicilios en corriente continua, se encargará de la alimentación de los motores del robot.



Figura 36: Fuente de alimentación

Condensador 100uF

En este caso, el condensador se conectará a la entrada de la tensión en el driver, funcionando como filtro para que no haya una variación de tensión a lo largo del tiempo.



Figura 37: Condensador para filtrado

Motor paso a paso Nema 17

El **motor paso a paso** (Stepper) conocido también como **motor** de pasos es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de girar una

cantidad de grados (**paso** o medio **paso**) dependiendo de sus
entradas de control.



Figura 38: Motor paso a paso
NEMA 17

Servomotor SG90

Los servomotores no son en realidad una clase específica de motor, sino una combinación de piezas específicas, que incluyen un motor de corriente continua o alterna, y son adecuados para su uso en un sistema de control de bucle cerrado. Se utilizan básicamente en la robótica industrial, en la fabricación con sistemas de automatización y en aplicaciones de mecanizado de control numérico (CNC) por ordenador.



Figura 39: Servomotor SG90

6.3 Software

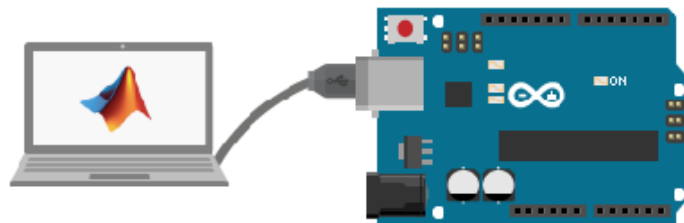
En este apartado se muestra como se ha configurado la conexión entre Matlab y arduino, la cuál se va a realizar a través del puerto serie del PC. Para poder realizar esta operación, es necesario descargar en Matlab el paquete de herramientas “MATLAB support for hardware Arduino” (FIGURA 7.3.1).

Name	Type	Author	Install Date	
 MATLAB Support Package for Arduino Hardware version 20.1.1	 Hardware Support Package		8 January 2021	 ⋮
 WLAN Toolbox version 3.0	 MathWorks Toolbox		23 October 2020	⋮
 Wireless HDL Toolbox version 2.0	 MathWorks Toolbox		23 October 2020	⋮

Figura 40: Paquete de Matlab para arduino.

Una vez se tiene el correspondiente paquete de herramientas, se puede establecer la conexión Matlab-arduino y configurar las distintas entradas/salidas del arduino según los requerimientos del usuario, a continuación, se muestra un ejemplo:

```
a = arduino('COM3','Uno','Libraries','Servo');
configurePin(a,'D6','DigitalOutput');
```



Donde:

a: Variable adjudicada por el usuario al arduino que se quiere configurar, mediante la función “arduino” se configura el puerto “COM3”, se selecciona el tipo de arduino “Uno”, se permite acceso a la carpeta “Libraries” con objetivo de utilizar la librería de “Servo”, especial para servomotores.

ConfigurePin: Esta función permite configurar cualquier pin “D6”, en el arduino “a”, como una salida digital.

7. MANUAL DE USUARIO

En este capítulo se muestran las instrucciones necesarias para el usuario con el objetivo de utilizar la aplicación de manera sencilla.

1º: En primer lugar, se encuentra la pantalla de inicio, donde se debe de realizar la selección de usuario y posteriormente podremos seleccionar la pantalla de TAREAS o la de ROBOT, la de simulación se encuentra inhabilitada ya que en el inicio no hay ningún programa cargado. (Figura 7.1)



Figura 41: Pantalla de Inicio

2º: Si el objetivo es realizar movimientos manualmente, se puede acceder a la pantalla de ROBOT, en este caso, el objetivo es programar una tarea, por lo tanto, el siguiente paso a realizar es pulsar el botón de tareas situado en la pantalla de INICIO. Una vez estemos dentro, podremos empezar a programar las tareas. (Figura 7.2).



Figura 42: Pantalla de Tareas

3º: Una vez se está en la pantalla de tareas, se pueden programar las tareas introduciendo los comandos deseados, para hacerlo correctamente hay que añadir cada movimiento ordenadamente, no pueden realizarse dos movimientos axiales simultáneamente, hay que elegir entre mover el eje x, z, girar base o cerrar pinza. En la siguiente figura se muestran los botones para introducir los comandos. (Figura 7.3)



Figura 43: Programación de tareas

4º: Tras introducir los comandos, hay que pulsar “convertir”, este comando activa las funciones que aproximan los valores teóricos a los valores reales que pueden ejecutar los motores del robot, tras convertir las instrucciones, se añaden en el programa pulsando el botón “añadir”.

5º: Una vez tenemos un programa de tareas definido, se pasa a la pantalla de SIMULACIÓN o ROBOT, donde podremos ejecutar el programa tanto en la simulación como en la realidad.

6º: Si se activa la pantalla de simulación, el programa de tareas es transferido a la aplicación de SIMULACIÓN, por lo tanto, se puede simular el programa, para ello hay que abrir RobotSim y ejecutar la simulación pulsando en “Simulación”. (Figura 7.5)



Figura 44: Pantalla de simulación.

7º: Si se activa la pantalla de ROBOT, lo primero que hay que hacer es configurar la conexión de Matlab-Arduino, para ello se introduce el puerto de conexión y el controlador deseado, posteriormente se pulsa en “Configurar” y ya estaría la conexión configurada.

8º: Llegados a este punto, es posible mover el robot manualmente, para ello, se introducen los parámetros deseados y se pulsa “Ejecutar”, por el contrario, si se prefiere una ejecución de un programa previamente programado, se pulsa “Ejecutar programa”. Antes de realizar estos pasos, es necesario pulsar “Posición inicial”, donde se indican los parámetros para posicionar el robot en la posición 0. (Figura 7.6)

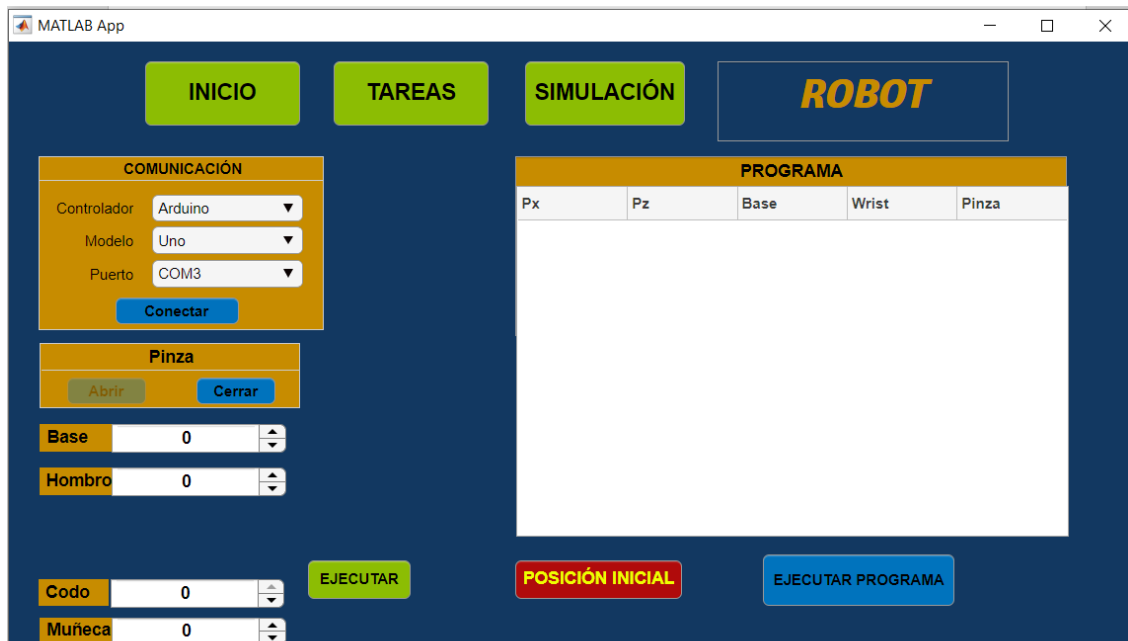


Figura 45: Pantalla de ROBOT.

9º: Llegados a este punto, se ha conseguido realizar todos los pasos correctamente, una vez ejecutadas las tareas deseadas, se procede a cerrar todas las aplicaciones.

8. PRESUPUESTO

En este apartado se presenta el presupuesto necesario para el desarrollo del robot Franuc 24.

8.1 Material

A continuación, se muestra en forma de tabla los costes asociados a la adquisición del material necesario para la elaboración del robot, por un lado se encuentra el material hardware necesario y por otro los dispositivos necesarios para la realización del proyecto.

Material el desarrollo del robot	Cantidad	Coste unitario [€/unidad]	Coste total [€]
PLA 1KG	1	16	16
Rodamientos 22mm/8mm	10	0.2	2
Rodamientos lineales LWL7B 8mm	2	1	2
Arduino UNO R3	1	6	6
Motores paso a paso Nema 17	4	5	20
Servomotor SG90	1	1	1
Cable DuPont 4P 50cm	6	0.8	4.8
Transformador AC/DC 230V/24V	1	5.2	5.2
Subtotal			57

Tabla 5: Presupuesto del material necesario para el desarrollo del robot.

Material la elaboración del proyecto	Concepto	Cantidad	Coste total [€]
Msi GF93	Portatil	1	850
Ender 3 Pro	Impresora 3D	1	183
Subtotal			933

Tabla 6: Presupuesto del material necesario para la elaboración del proyecto

8.2 Licencias de software

A continuación, se muestran los costes de las diversas licencias de software empleadas durante el desarrollo del estudio.

Licencia de software	Cantidad	Coste unitario [€/unidad]	Coste total [€]
SolidWorks 2018 x64	1	Gratis para estudiantes	0
Matlab	1	69	69
Ultimaker Cura	1	Gratis	0
Subtotal			69

Tabla 7: Presupuesto de las licencias de software necesarias para el desarrollo del proyecto

8.2 Presupuesto final

A continuación, se muestran los costes de las diversas licencias de software empleadas durante el desarrollo del estudio.

Tipo de coste	Coste total
Material el desarrollo del robot	57

Material la elaboración del proyecto	933
Licencia de software	69
Subtotal	1059

Tabla 8: Presupuesto final

9. CONCLUSIONES Y TRABAJOS FUTUROS

Finalmente, se ha llegado a este punto, en el que se valora el trabajo realizado y se cuestiona si los objetivos han sido alcanzados, para posteriormente, finalizar la memoria, si se vuelve a las primeras páginas, encontramos el capítulo 1: Motivación, objetivos y estructura, se puede hacer un pequeño repaso de los objetivos marcados en la estructura se puede comprobar que se ha conseguido completar los objetivos:

- Diseño de nuestro robot “Franuc 24” en Solidworks.
- Creación de un gemelo digital (DT), una realidad virtual de nuestro robot mediante Matlab, a través de sus herramientas “Simulink Simscape” y “simulink 3D animation”.
- Realización del estudio de la cinemática directa e inversa, así como la dinámica de nuestro robot.
- Creación de una GUI (interfaz gráfica de usuario) mediante AppDesigner de Matlab.
- Creación de la interfaz Matlab-Arduino para el control de los motores paso a paso.
- Diseño del control de potencia y una versión imprimible del robot “Franuc 24”.
- Elaboración de un protocolo de utilización adecuado de nuestro robot.
- Puesta en marcha del robot.

Más allá de los objetivos cumplidos en lo que a estructura se refiere, se procede a la valoración del trabajo realizado durante el proyecto, en primer lugar,

mentonar que uno de los principales objetivos era la sincronización del robot físico con su gemelo digital, esto no ha podido ser posible ya que el límite de este objetivo lo marca Matlab, el cual no permite una ejecución de programa simultanea, sino secuencial, por lo tanto no se ha podido realizar una correcta sincronización de los motores, esto ha generado problemas con la sincronización de los motores de la simulación, implicando, que de esta manera, se pueda realizar un control exacto de la velocidad y posición de los motores. No obstante, se ha conseguido realizar una aplicación que permita la simulación del control de posición del robot y el control de posición del robot físico a través de la aplicación.

En lo que se refiere al tema de la impresión, se han modificado algunos parámetros que venían predeterminados en la impresora, lo cual ha hecho posible una impresión rápida y aceptable, como imprimir a 40mm/s y con un relleno del 10%, anteriormente las impresiones se realizaban a 20mm/s y relleno del 40%, ocasionando retrasos de hasta 24 horas en algunas piezas y desaprovechando material.

Para finalizar la conclusión, mencionar la importancia de realizar un estudio de las fuerzas que sufren los motores debido a las fuerzas a las que son sometidos, ya que en caso de no hacerlo, es probable que los motores no estén bien dimensionados. En este proyecto no se ha realizado el estudio previamente dicho y se han probado los motores mediante prueba-error, esto ha provocado tener que comprobar si los motores son capaces de soportar las necesidades del Robot "Franuc 24". La falta de este estudio, viene ocasionado debido a que a la parte de diseño del robot físico no se le ha dedicado el tiempo necesario, el 80% del tiempo ha sido dedicado a la realización del gemelo digital, desarrollo de la aplicación, estudio cinemático del robot y los límites del mismo.

Para las futuras versiones del robot "Franuc 24", se presentan una serie de mejoras que se han ido gestando a lo largo de este proyecto, así como evitar errores y sus posibles mejoras:

- En este robot la posición de las articulaciones se realiza a través de unos indicadores impresos en las articulaciones del robot. La posible solución para saber exactamente el ángulo en el que se encuentra el motor sería mediante la implementación de potenciómetros, estos potenciómetros se pondrían en el eje de las articulaciones permitiendo conocer la posición exacta. Previamente habría que sacar la relación entre Grados/Ohmios para poder saber de una manera exacta los grados.
- Para conseguir una rotación y una fijación en la base más adecuada, se podría implementar un rodamiento que tenga un diámetro similar a la base, de esta manera conseguiríamos un giro sin apenas rozamiento y con una buena fijación a la base.
- Implementación de un sistema motriz adecuado a las necesidades del robot, se puede ver que en este diseño la implementación de motores con mayor par de fuerza hubiese facilitado la realización de unos movimientos totalmente correctos, en el diseño actual el momento que ejerce en las articulaciones impide el correcto funcionamiento del brazo robótico.
- Implementación de los micropasos en los motores paso a paso, evitar tener errores de posición, se pueden activar los pines de micropasos que los drivers disponen, con ello, se podría obtener una mayor resolución de los motores.
- Optimización en el tamaño del robot, estudio de las fuerzas que ejercen en las distintas articulaciones e implementación de un sistema de engranajes optimizado.
- Implementación de la articulación de la muñeca, en esta versión del Franuc 24 se ha suprimido el uso del motor situado en la muñeca debido a que carecía de sentido su utilización, el objetivo de un motor en la muñeca es el de rectificar los movimientos realizados por la base, para así tener el TCP en un eje deseado, objetivo que no se cumplía, por lo tanto, se propone el objetivo de la creación de una articulación en la muñeca que se ajuste a los requerimientos anteriormente dichos.

- Respecto a la simulación, implementación de un algoritmo que permita simular el robot tanto en velocidad como en posición, en esta primera versión, sólo es posible una simulación del robot en términos de posición, debido a que no se pueden ejecutar simultáneamente las simulaciones de las distintas articulaciones.

ANEXOS

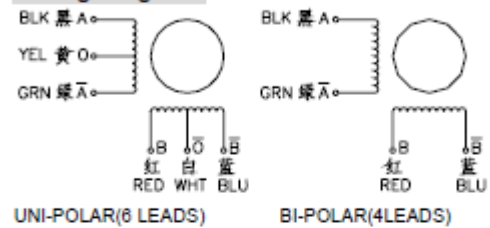
ANEXO I: DATASHEETS



HB Stepper Motor Catalog

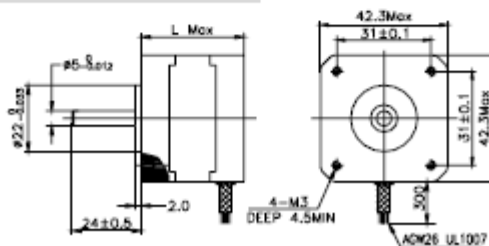
MotionKing (China) Motor Industry Co., Ltd.

2 Phase Hybrid Stepper Motor 17HS series-Size 42mm(1.8 degree)

**Wiring Diagram:****Electrical Specifications:**

Series Model	Step Angle (deg)	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm ²)	Lead Wire (No.)	Motor Weight (g)
17HS2408	1.8	28	0.6	8	10	12	1.6	34	4	150
17HS3401	1.8	34	1.3	2.4	2.8	28	1.6	34	4	220
17HS3410	1.8	34	1.7	1.2	1.8	28	1.6	34	4	220
17HS3430	1.8	34	0.4	30	35	28	1.6	34	4	220
17HS3630	1.8	34	0.4	30	18	21	1.6	34	6	220
17HS3616	1.8	34	0.16	75	40	14	1.6	34	6	220
17HS4401	1.8	40	1.7	1.5	2.8	40	2.2	54	4	280
17HS4402	1.8	40	1.3	2.5	5.0	40	2.2	54	4	280
17HS4602	1.8	40	1.2	3.2	2.8	28	2.2	54	6	280
17HS4630	1.8	40	0.4	30	28	28	2.2	54	6	280
17HS8401	1.8	48	1.7	1.8	3.2	52	2.6	68	4	350
17HS8402	1.8	48	1.3	3.2	5.5	52	2.6	68	4	350
17HS8403	1.8	48	2.3	1.2	1.6	46	2.6	68	4	350
17HS8630	1.8	48	0.4	30	38	34	2.6	68	6	350

*Note: We can manufacture products according to customer's requirements.

Dimensions: unit=mm**Motor Length:**

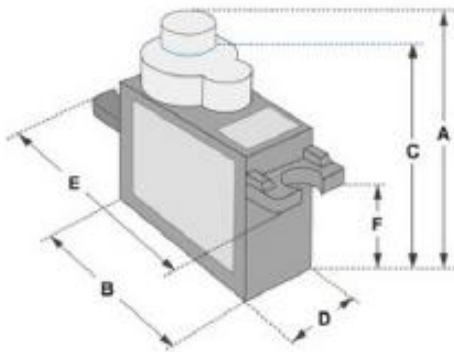
Model	Length
17HS2XXX	28 mm
17HS3XXX	34 mm
16HS4XXX	40 mm
16HS8XXX	48 mm

SERVO MOTOR SG90

DATA SHEET



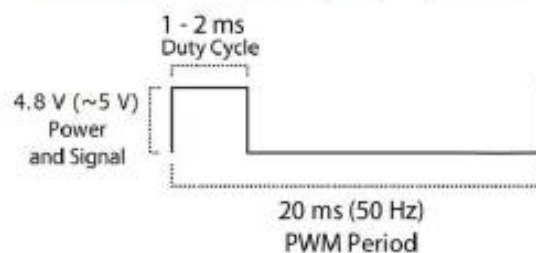
Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Dimensions & Specifications	
A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

PWM=Orange (⏏)
Vcc=Red (+)
Ground=Brown (-)

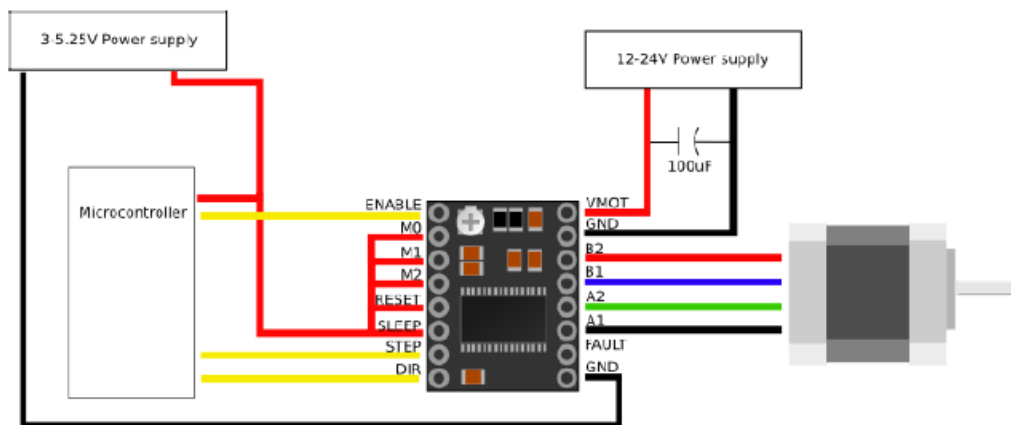


STEPSTICK DRV8825 v1.0 DATASHEET



EXAMPLE CIRCUIT

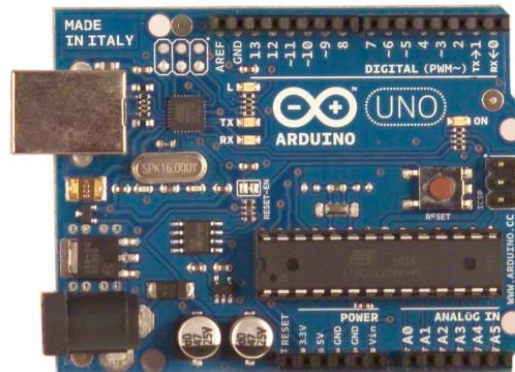
The following diagram lists the pins and show an example circuit.



Name	Description
Enable	Enable/disable the stepper driver HIGH - Disable LOW - Enable *
M0 - M2	Step resolution setting, see chapter 'step resolution configuration'
RESET	Enable/disable the H-bridge output * LOW - Disable * HIGH - Enable
SLEEP	Enable/disable low-power sleep mode LOW - Sleep * HIGH - Active
STEP	LOW → HIGH, move one step
DIR	LOW / HIGH switches direction
VMOT	Motor power (12-24V)
GND	System ground
FAULT	LOW when the stepper driver is in fault condition. You can provide 5V on this pin for compatibility with stepstick A4988

* this is the default state when the pin is not connected

Arduino UNO



CE

Technical Specification

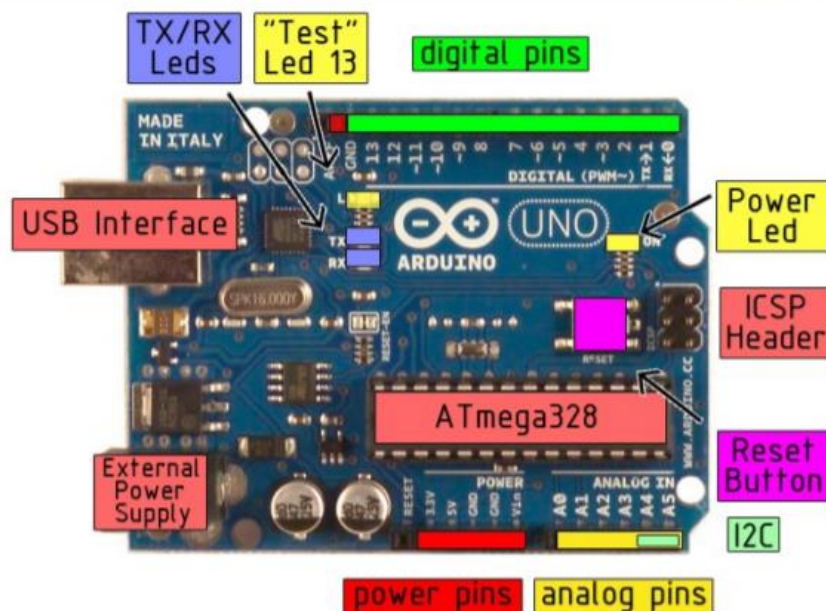


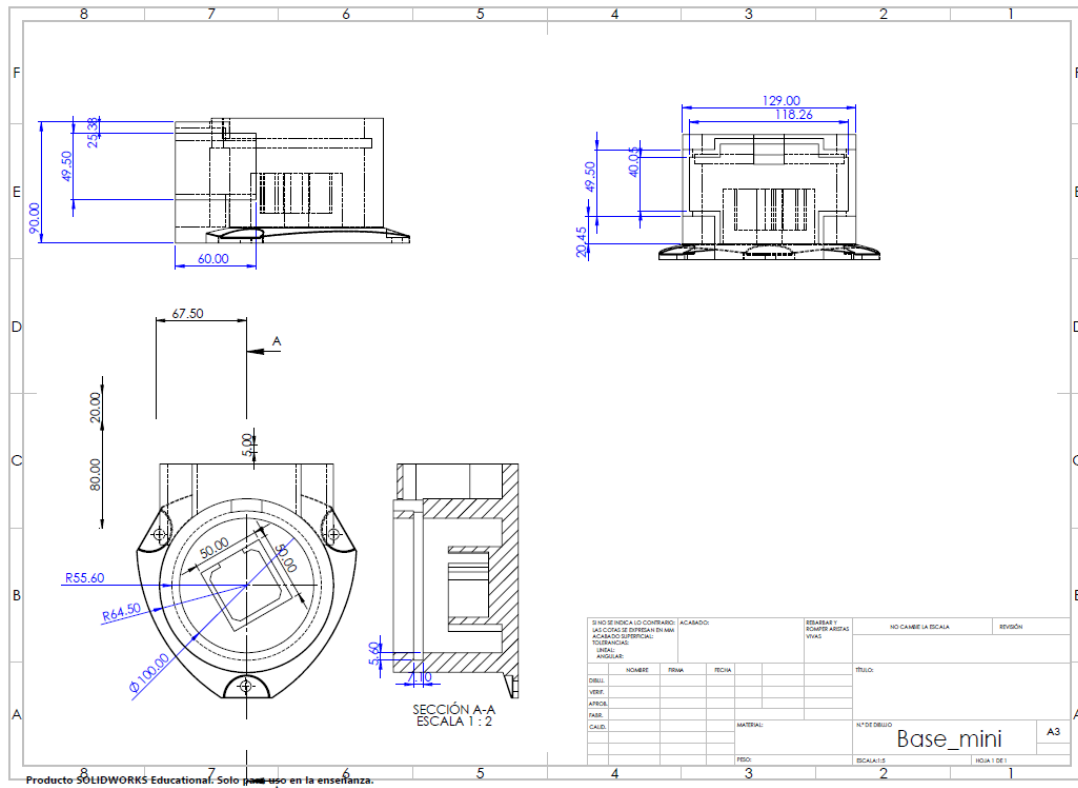
EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

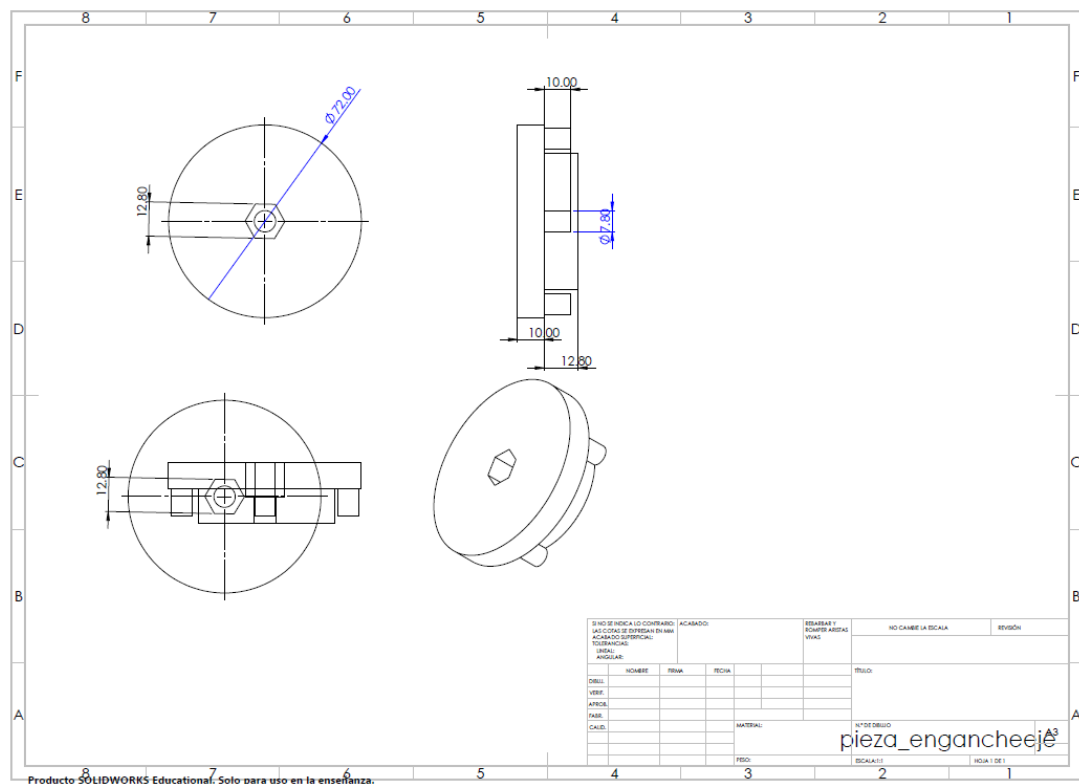
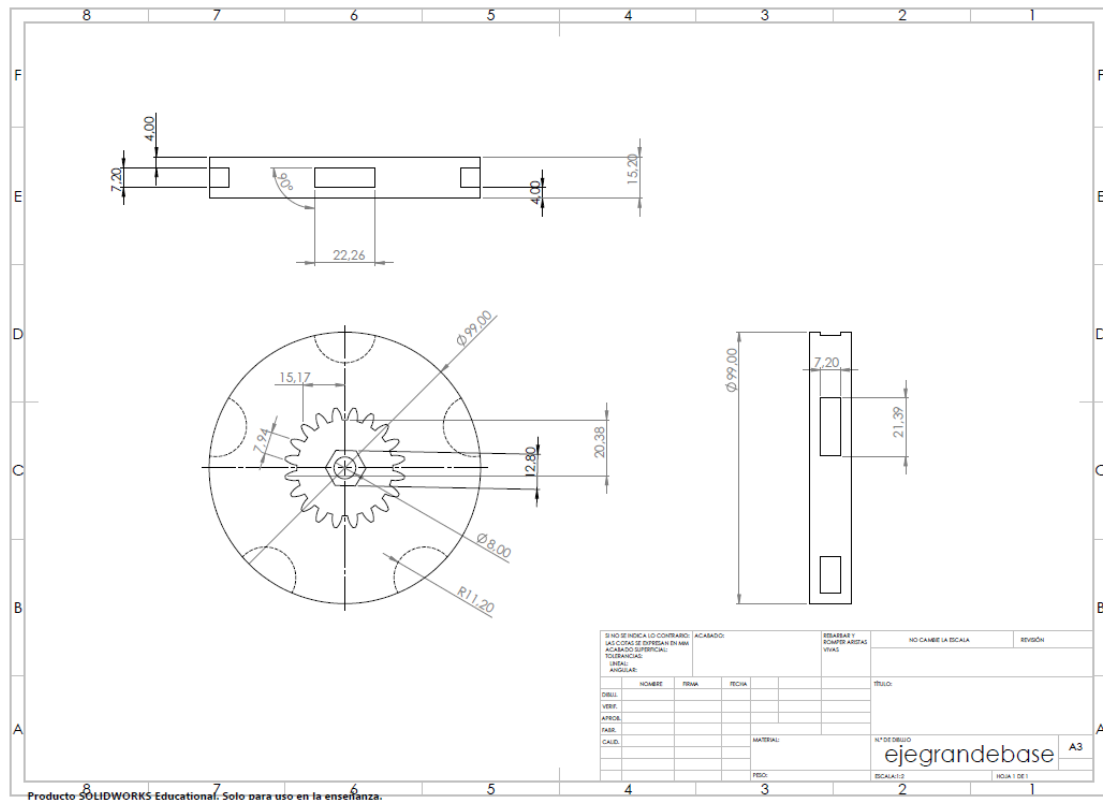
Summary

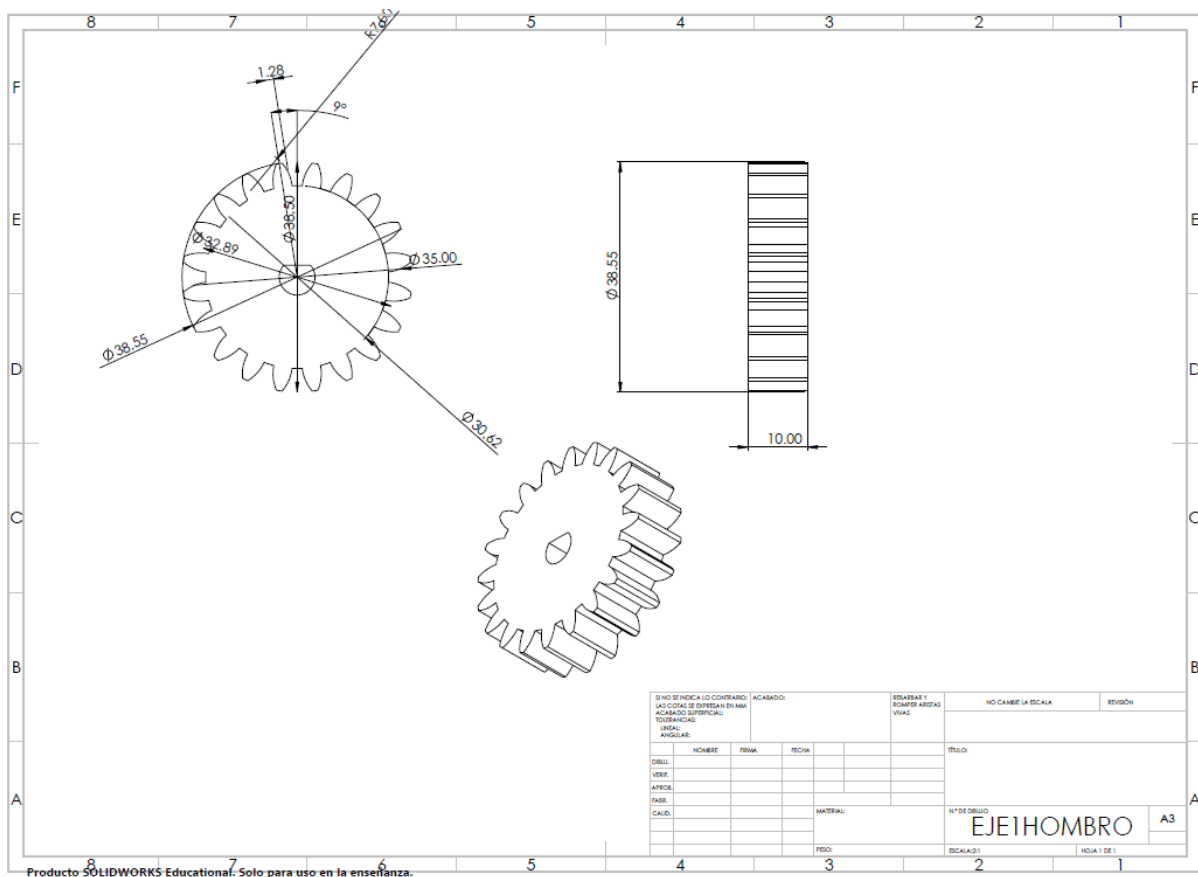
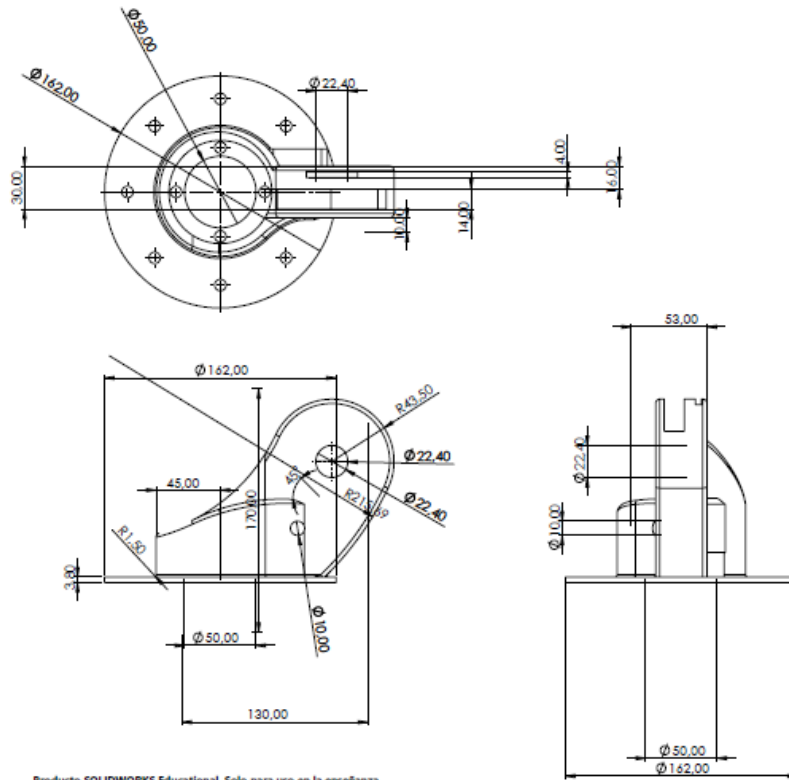
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

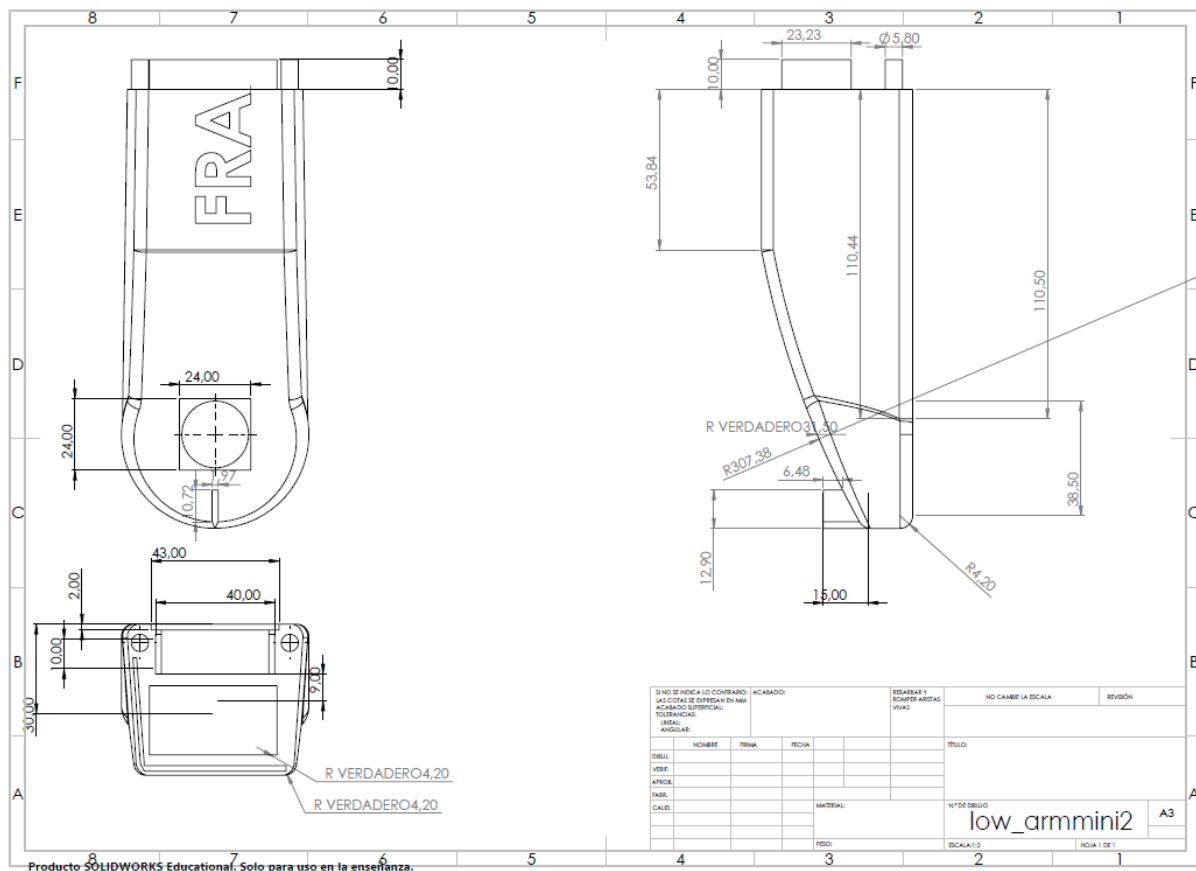
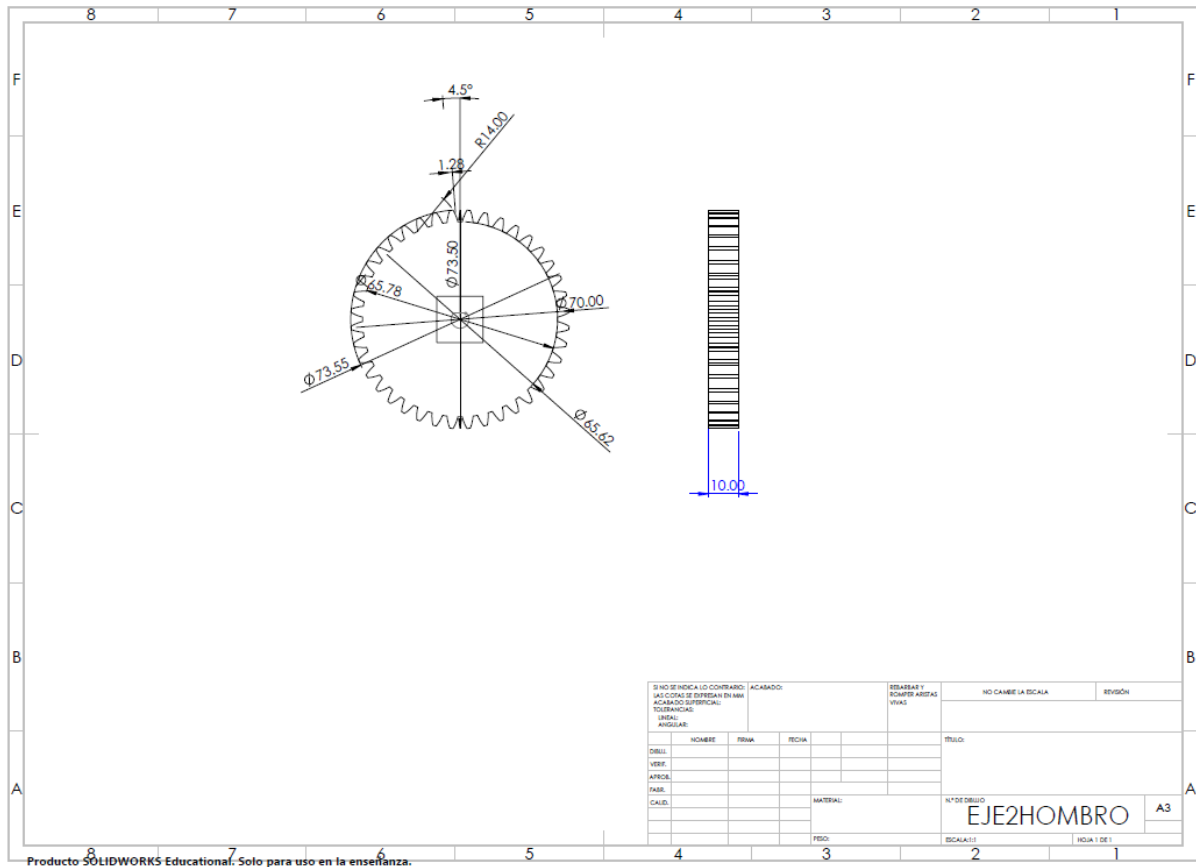
the board

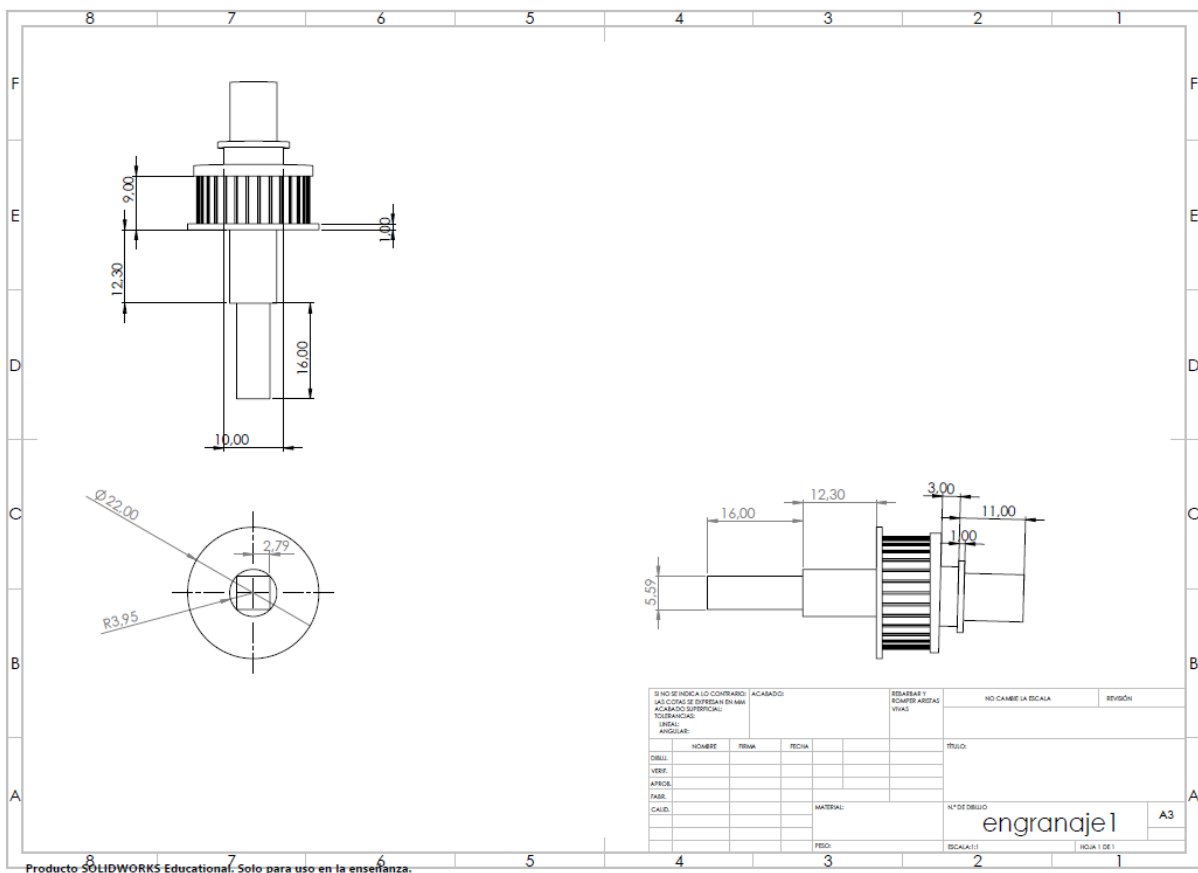
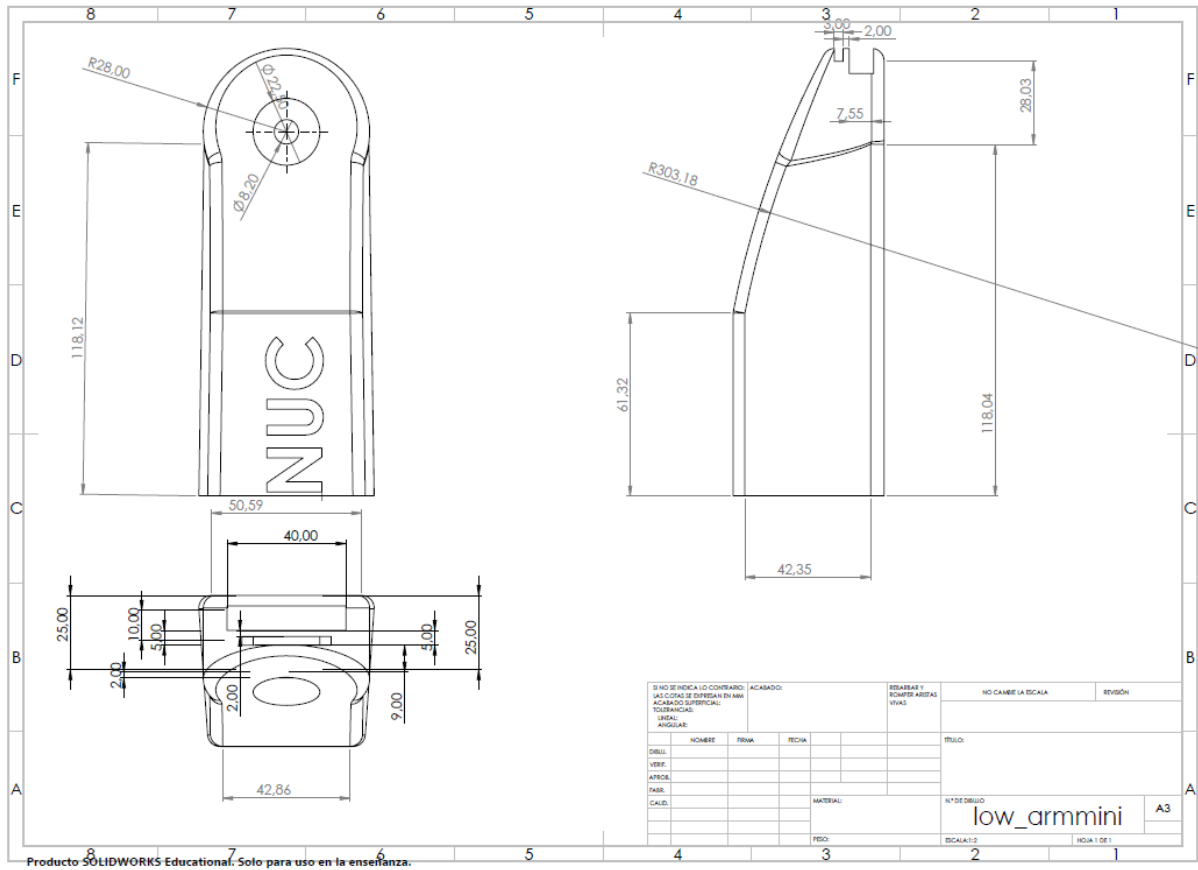


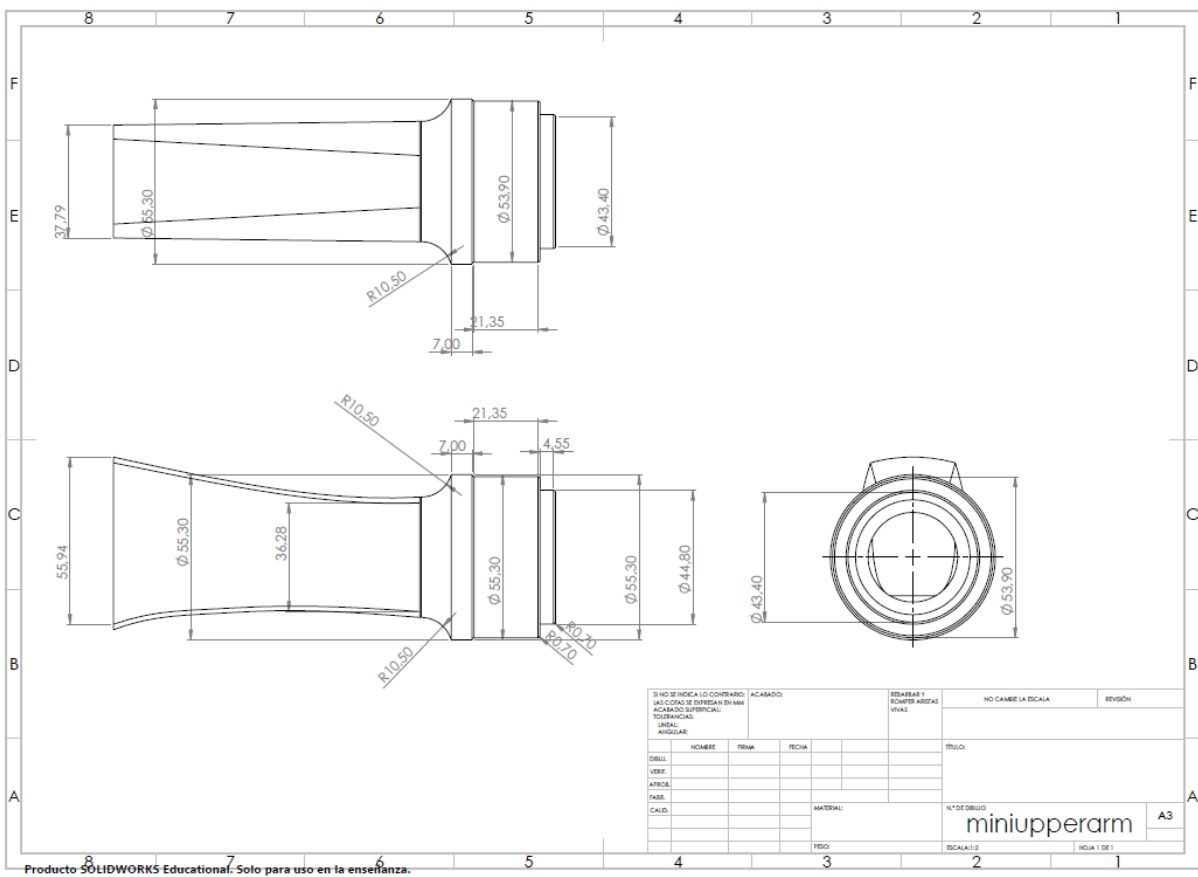
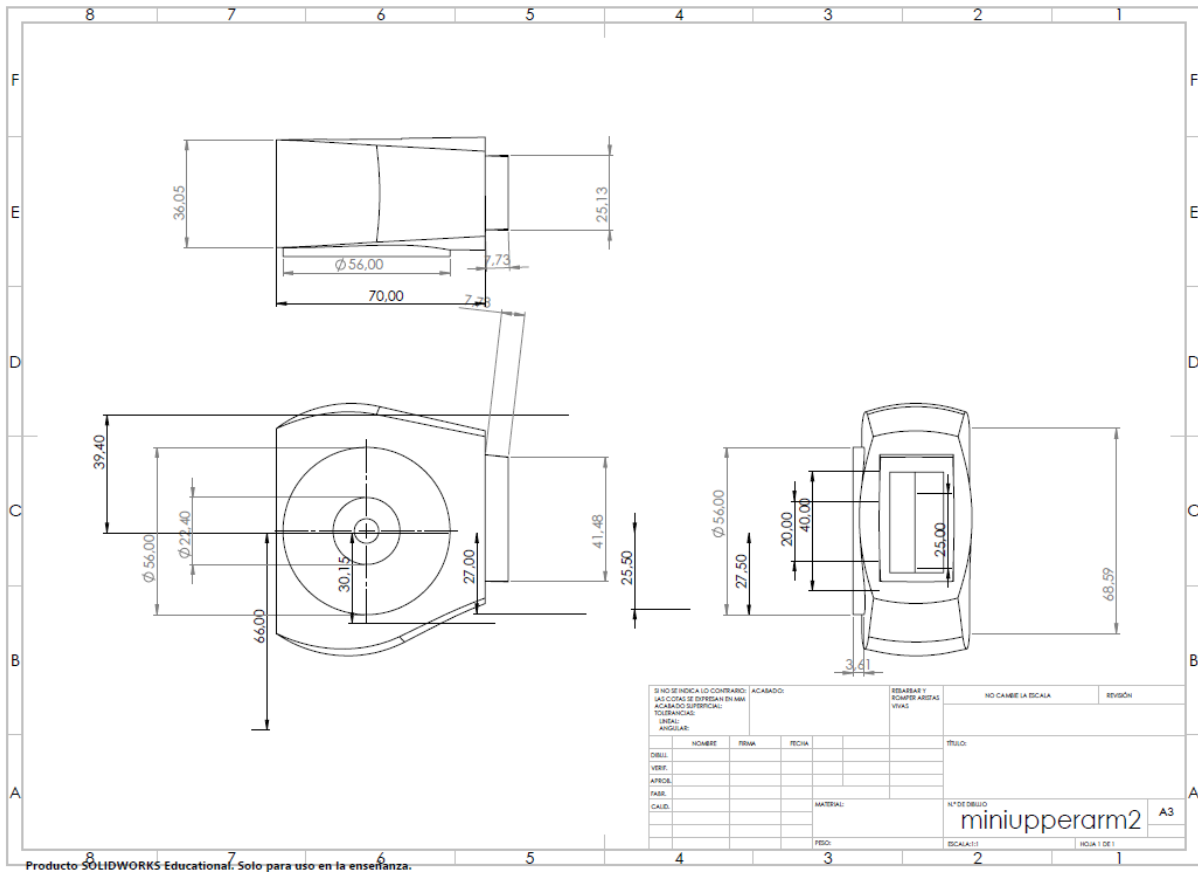


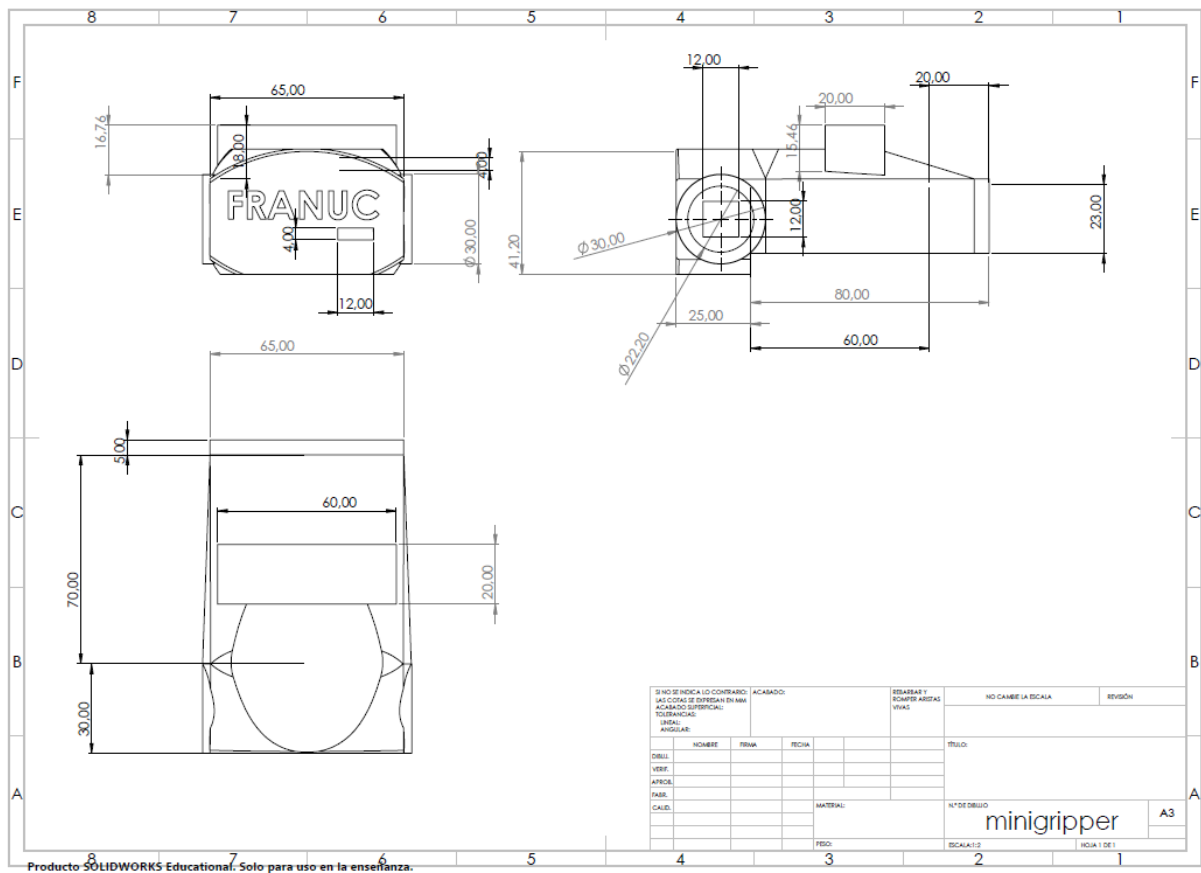
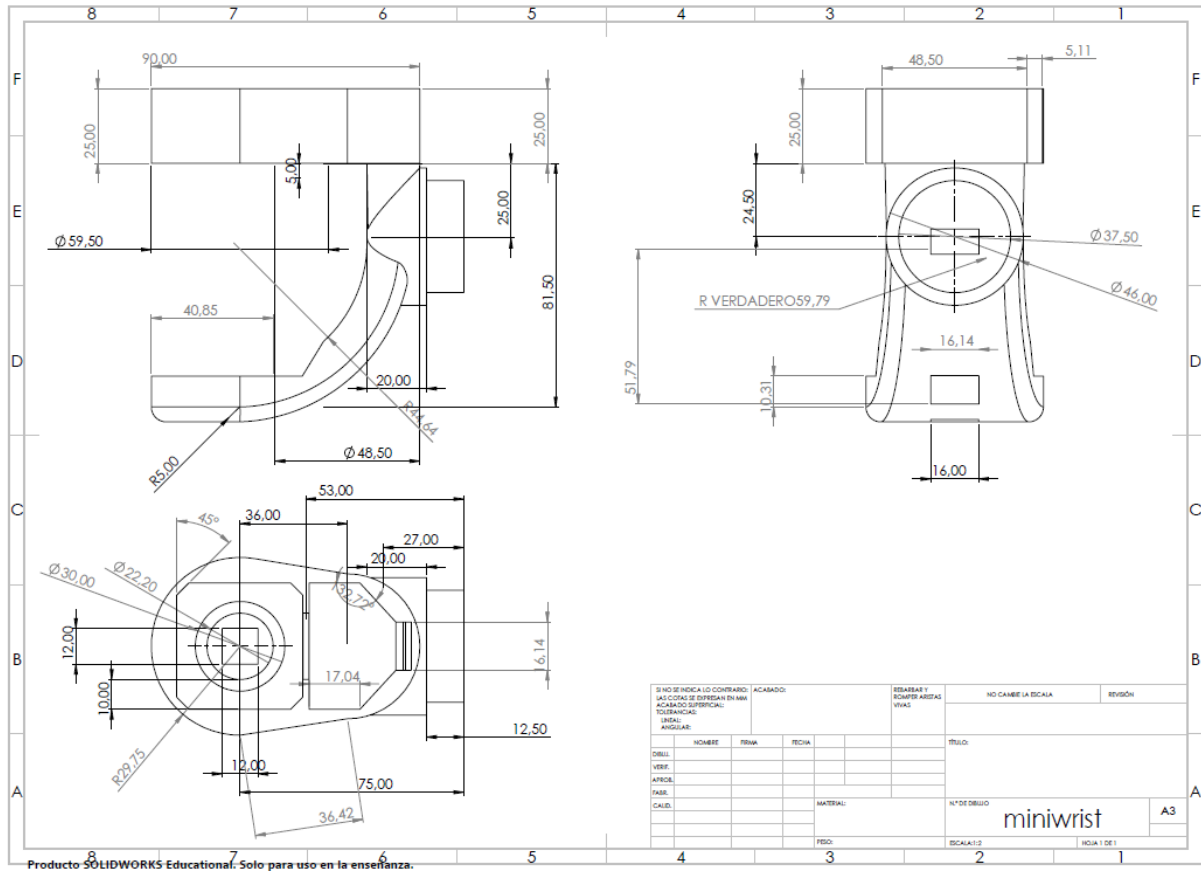


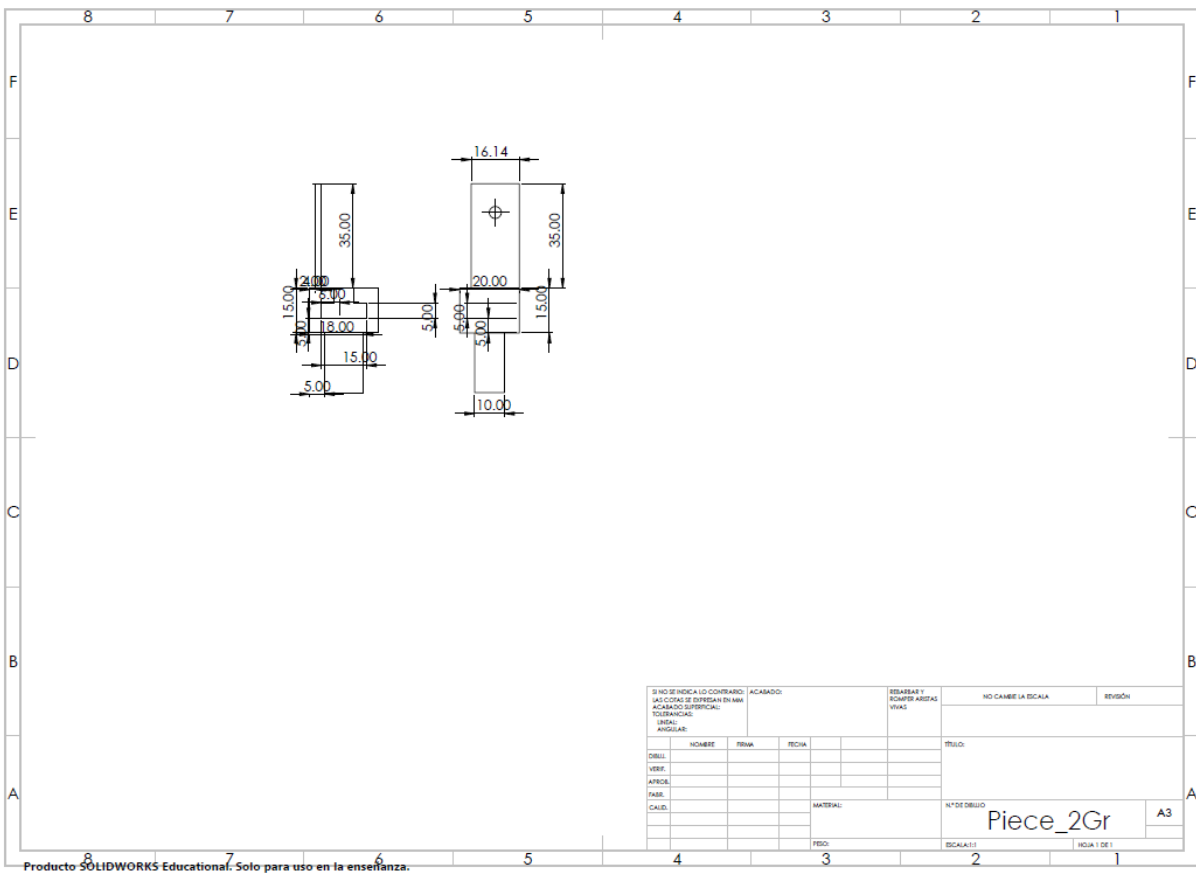
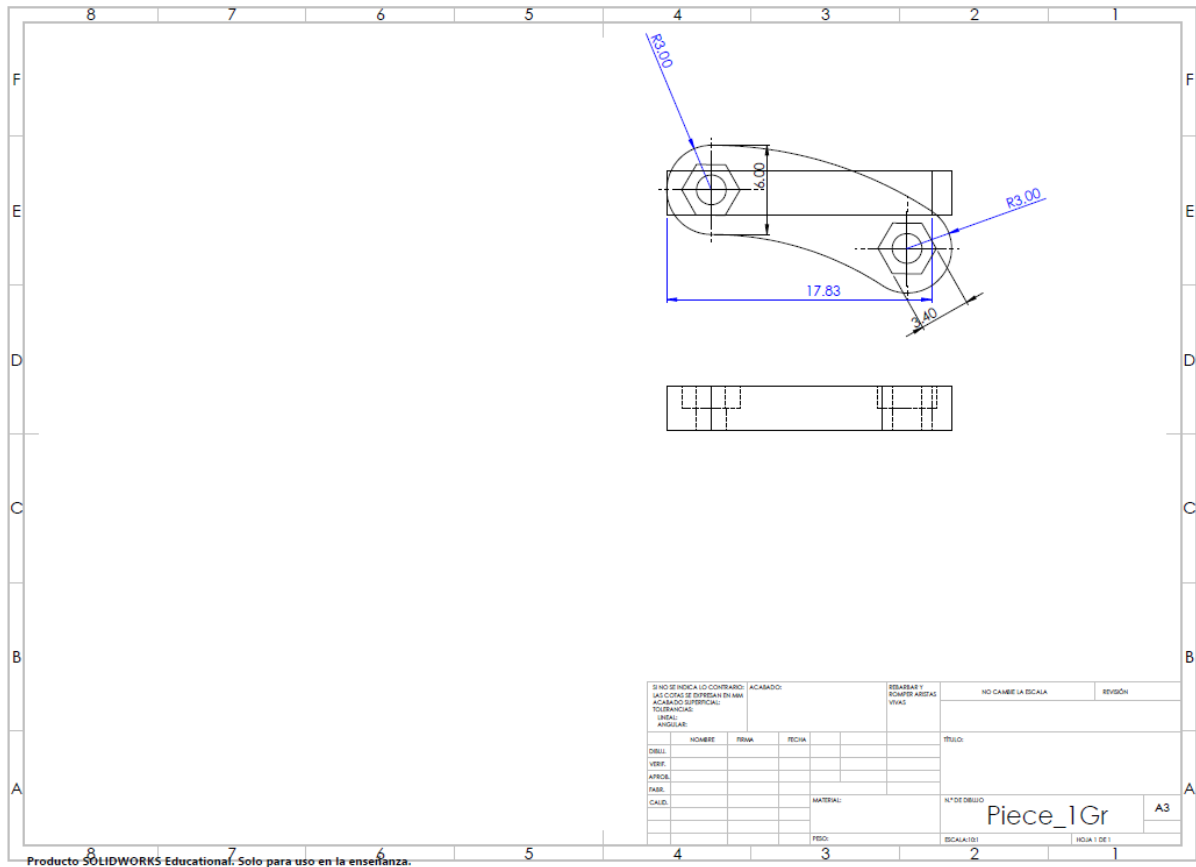


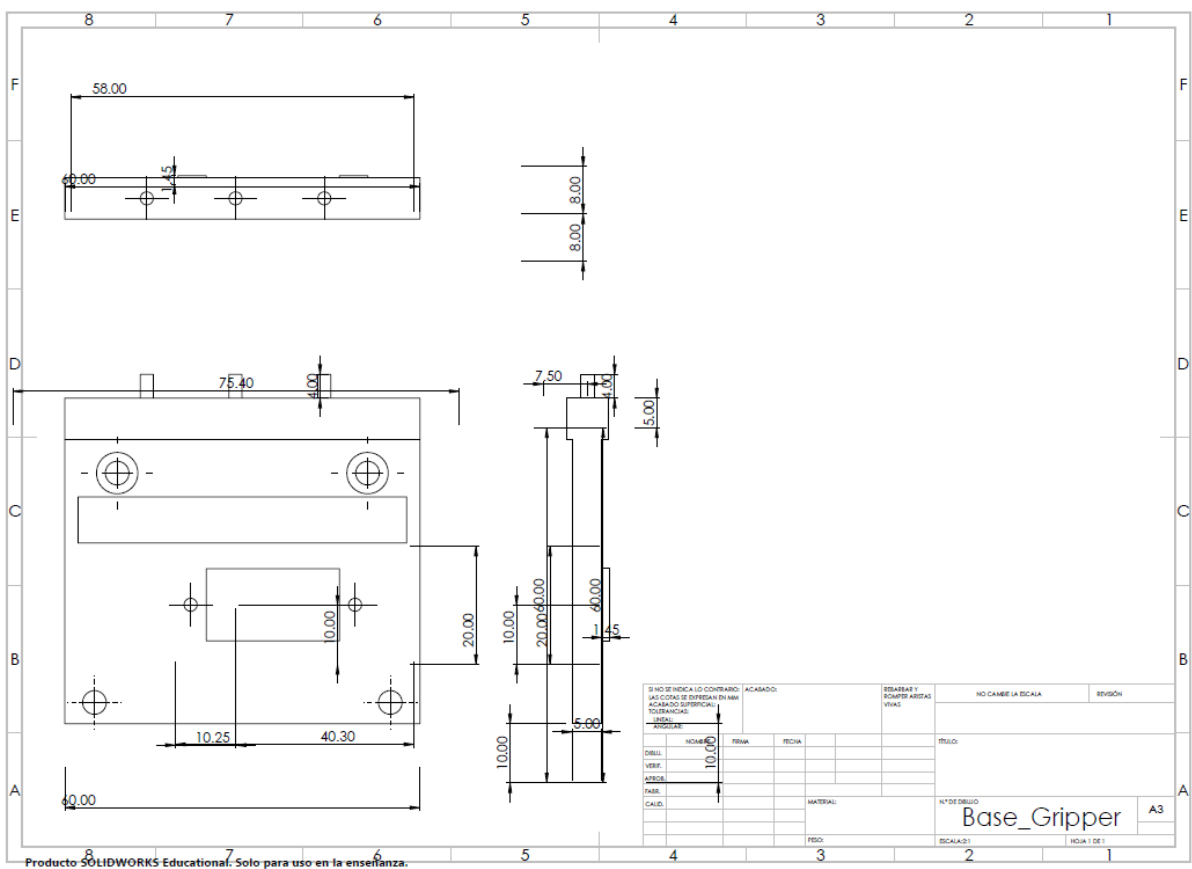
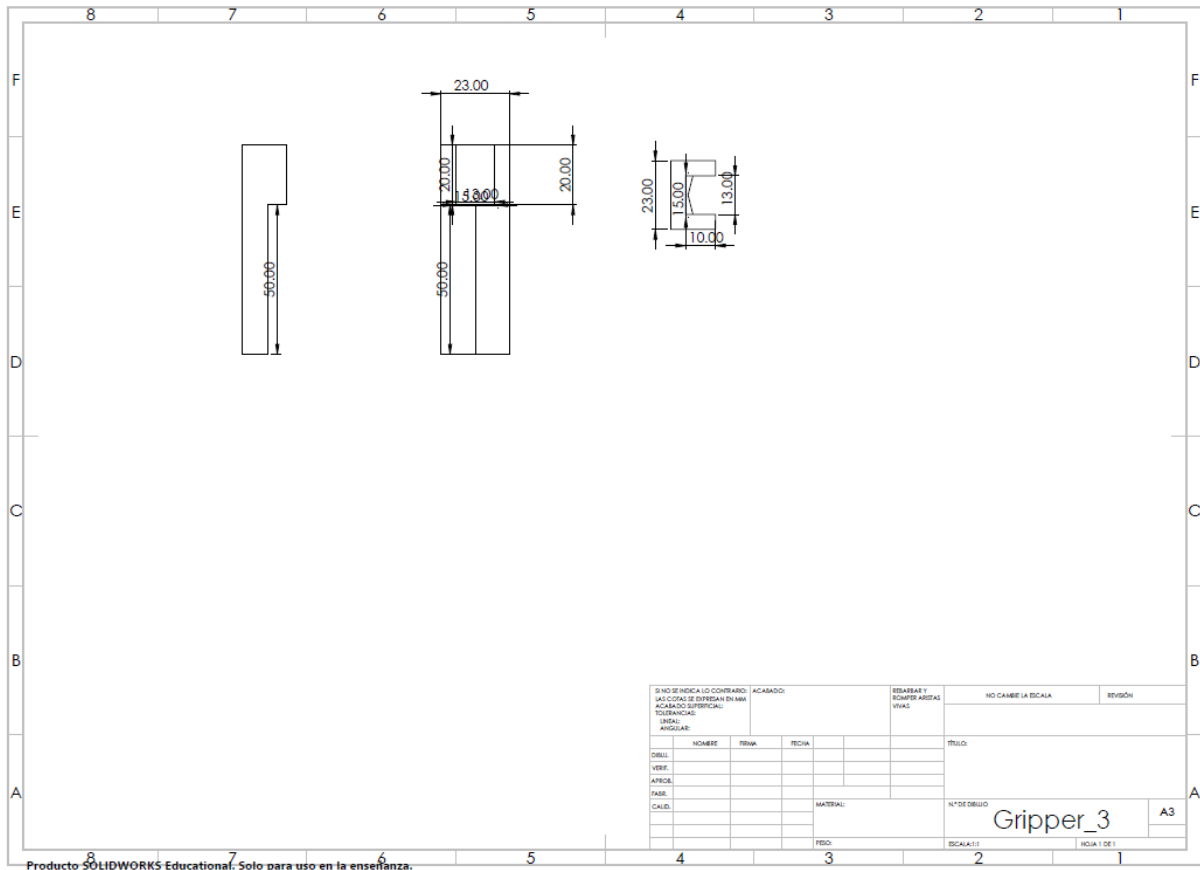












Bibliografía

Creación de gemelo digital

https://es.mathworks.com/help/sl3d/link-to-simulink-and-simmechanics-models.html?searchHighlight=simulink%20simscape&s_tid=srchtitle

Creación de aplicación en AppDesigner

https://es.mathworks.com/help/matlab/app-designer.html?searchHighlight=app%20designer&s_tid=srchtitle

Ayuda general MATLAB

https://es.mathworks.com/?s_tid=gn_logo

Comunicación MATLAB-Arduino

<https://geekytheory.com/matlab-arduino-serial-port-communication>

Diseño Robot “Franuc 24”

https://www.youtube.com/channel/UCX6c67E5PTUq5W8n8POeO_w/featured

Impresión 3D

<http://imprimalia3d.com/recursosimpresion3d/gu-configuraci-n-par-metros-slic3r>

Control de motores paso a paso

<http://www.dima3d.com/motores-paso-a-paso-en-impresion-3d-ii-criterios-de-seleccion-de-motores-y-drivers/>