

Documentación para uso del Simulador **ALMarDrawingRobot**

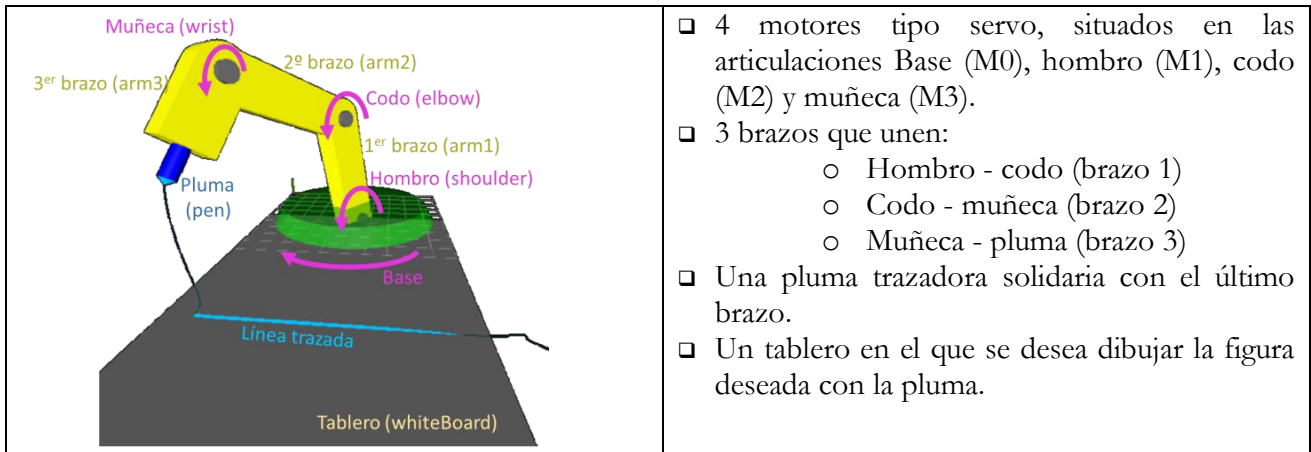
Indice

1.	Introducción.....	1
2.	Descargas e instalaciones necesarias	1
2.1.	Simulador genérico.....	1
2.1.	Plugin ALMarDrawingRobot para simulador genérico.....	1
3.	Ejecución del simulador	1
3.1.	Espacio de trabajo	2
3.2.	Sistemas de coordenadas e interacción desde consola	2
4.	Control de los servos desde programa de usuario	3
4.1.	Librerías para C/C++	3
4.1.1.	Compilación con la librería UserLibSimulator	3
4.1.1.	Compilación con la librería UserLibALMarDuino	4
5.	Interacción con el simulador desde Matlab/Simulink	4
6.	Interacción con el simulador desde Python.....	4

1. Introducción

El simulador **AlMarDrawingRobot**, junto con las herramientas de desarrollo adecuadas, permite diseñar, realizar y analizar el control de un brazo robótico con 4 grados de libertad, con movimientos realizados mediante servomotores. El sistema tiene un funcionamiento similar al robot educativo Tinkerkit Braccio (<https://store.arduino.cc/products/tinkerkit-braccio-robot>).

El sistema simulado está formado por:



Mediante el accionamiento de los servos, el usuario podrá llevar la pluma trazadora a la posición del tablero y generar los trazos deseados sobre el mismo.

2. Descargas e instalaciones necesarias

Se necesitan los programas siguientes para el uso del simulador.

2.1. Simulador genérico

El simulador base se encuentra disponible para su uso desde un equipo Windows 10 o superior, Linux o Mac en el enlace <http://isa.uniovi.es/~ialvarez/Curso/descargas/SimuladorDocker/dockerimagedistrib.zip>.

Las instrucciones para su instalación y uso del simulador genérico se encuentran en el enlace <http://isa.uniovi.es/~ialvarez/Curso/descargas/SimuladorDocker/almar-simulator-install.pdf>. No es necesario reinstalar el simulador genérico ni sus librerías cliente si ya se dispone del mismo para otras aplicaciones.

2.1. Plugin AlMarDrawingRobot para simulador genérico

Para añadir el plugin requerido del robot, se deben realizar los pasos siguientes:

- ❑ Descargar el plugin específico para su uso con el simulador, desde el enlace <http://isa.uniovi.es/~ialvarez/Curso/descargas/SimuladorDocker/AlMarDrawingRobot.zip>.
- ❑ Descomprimir el archivo obtenido en la carpeta **Server** del directorio donde se ha instalado el simulador, siguiendo las instrucciones de la instalación del simulador genérico.

Una vez realizados los pasos anteriores, ya se puede comprobar el funcionamiento del simulador (ver punto 3).

3. Ejecución del simulador

Tras la ejecución del simulador (ver instrucciones en `almar-simulator-install.pdf`), la visualización en tiempo real del estado se realiza conectando un cliente web con la dirección: <http://127.0.0.1:8080>. Solicitará nombre de usuario y contraseña: `alumno/ISAUNIOVI`. A continuación, se seleccionan la aplicación **AlMarDrawingRobot** a través de su icono, y cuando se observe el robot en su espacio de navegación (en la primera activación tarda unos segundos porque tiene que cargar todos los archivos

necesarios, en las siguientes será más rápido) se pulsa el botón **Run** para que el simulador comience a funcionar.

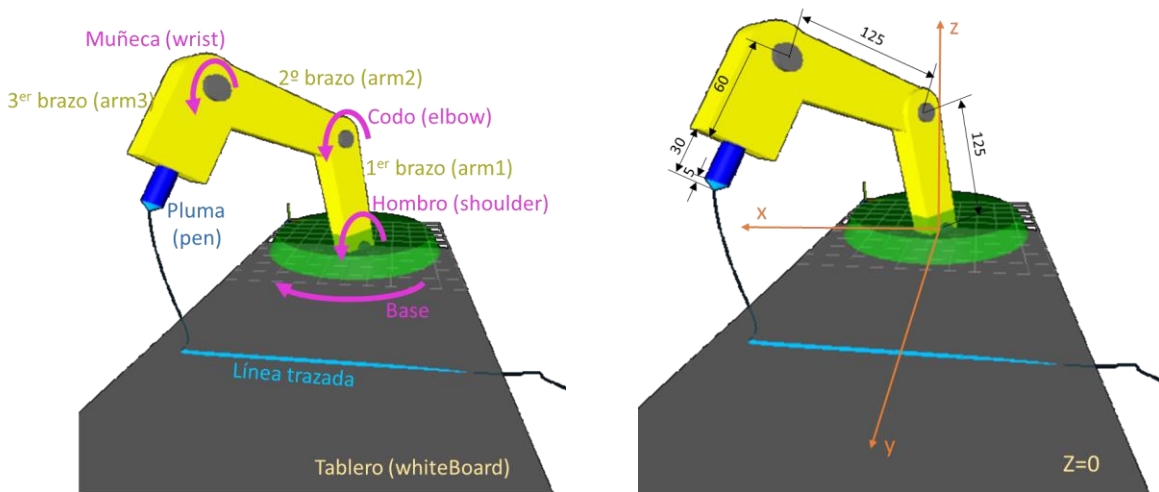
En caso de problemas la página puede recargarse (pulsar botón Actualizar en el navegador), lo que revierte el sistema a su posición inicial. En general, no es necesario detener el funcionamiento ni actualizar hasta el final de la sesión de trabajo.

El simulador se quedará ejecutando en la ventana Docker. Para terminarlo, seguir los pasos indicados en almar-simulator-install.pdf. Si no se hace esto, la aplicación se mantendrá consumiendo recursos (memoria, tiempo de procesador, batería).

La interacción del usuario con el simulador se realiza a través de la gestión de las señales PWM, y mediante la activación del pulsador CLR en el interfaz (borrar dibujo).

3.1. Espacio de trabajo

Las figuras siguientes muestran los diferentes elementos y sus dimensiones, así como el sistema de coordenadas **xyz** utilizado para el robot. Las articulaciones base y hombro se encuentran situadas en el origen de coordenadas 0,0,0. El tablero se encuentra en el plano $Z=0$.



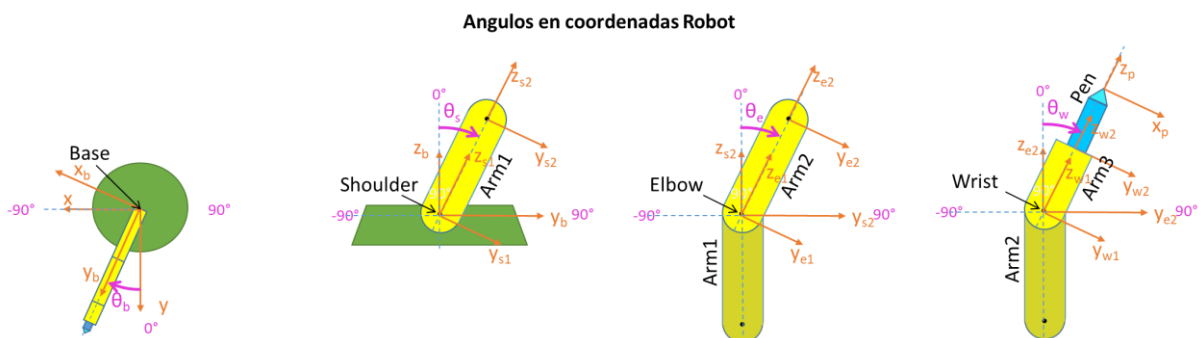
Para dibujar correctamente, la pluma debe mantenerse vertical respecto al tablero.

Las articulaciones están numeradas de 0 a 3 (base, hombro, codo, muñeca).

3.2. Sistemas de coordenadas e interacción desde consola

En la página web se dispone de una consola, en la que se pueden introducir comandos básicos para movimiento del robot, que comenzarán por el carácter # :

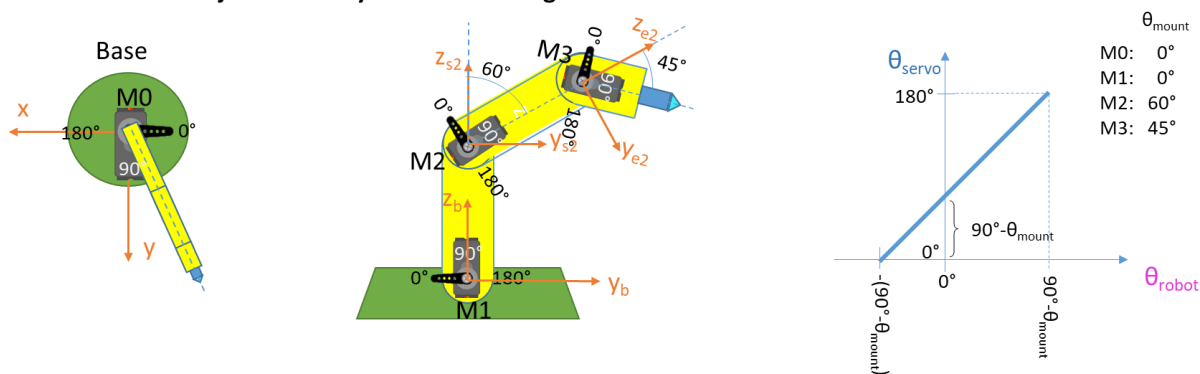
- ❑ **#d[ind articulación 0 a 3]=ángulo en grados robot:** gira la articulación deseada a la posición absoluta indicada (en grados de robot). Los ángulos se interpretan según la figura siguiente, que se resume así: 0° en la base corresponde al eje Y, 0° en el hombro se corresponde con posición vertical, 0° en el resto hace que los brazos que una queden alineados.



- ❑ **#s[ind articulación 0 a 3]=ángulo en grados servo:** gira la articulación deseada a la posición absoluta indicada (en grados de servo). Todos los servos tienen un rango 0° a 180° .

Los ángulos se interpretan según la figura siguiente, ya que cada servo se ha instalado de forma que su rango sea compatible con los movimientos deseados del robot.

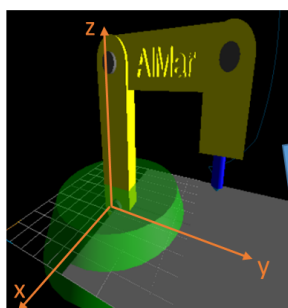
Montaje de servos y relación con ángulos en coordenadas Robot



- #p[ind articulación θ a 3]=valor en ms: activa el servo deseado mediante la generación de la señal PWM indicada. Todos los servos utilizan una PWM de periodo 20ms, en el cual la relación entre tiempo on y ángulo de giro servo es la siguiente:

- 1ms Ton \leftrightarrow 0° servo
- 1.5ms Ton \leftrightarrow 90° servo
- 2ms Ton \leftrightarrow 180° servo

A modo de ejemplo: las 3 versiones de comandos siguientes producen la misma posición de las articulaciones que se aprecia en la figura.



	Giro base	Giro hombro	Giro codo	Giro muñeca
Por ángulos robot:	#d[0]=0	#d[1]=0	#d[2]=90	#d[3]=90
Por ángulos servo:	#s[0]=90	#s[1]=90	#s[2]=120	#s[3]=135
Por Ton servo (ms):	#p[0]=1.5	#p[1]=1.5	#p[2]=1.667	#p[3]=1.75

4. Control de los servos desde programa de usuario

El control de los servos puede realizarse desde C/C++, Python, Matlab o, en general, cualquier programa que permita una comunicación websocket. Para los anteriores se dispone de librerías que facilitan el desarrollo.

Para conectar con el simulador, se dispone de un checkbox en la ventana de visualización web, que indica los valores (usuario, clave, etc.) a introducir para la conexión.

4.1. Librerías para C/C++

Se dispone de las librerías **UserLibSimulator** y **UserLibAlMarDuino** para control de los servos desde C/C++.

- La librería **UserLibSimulator** permite un control total de los servos desde una aplicación C/C++ modo consola, con UI, etc.
- La librería **UserLibAlMarDuino** utiliza la anterior pero permite un interfaz de la aplicación estilo Arduino (sólo C++ modo consola), con `setup()`, `loop()`, y uso de la clase `Servo`.

4.1.1. Compilación con la librería UserLibSimulator

Para el uso de la librería **UserLibSimulator** se dispone de ayuda on-line accesible con doble-click sobre el archivo `Client/help.html` de la carpeta de descarga. A partir de dicho enlace, mediante el

navegador se puede acceder a toda la documentación para el uso de esta librería desde C/C++; la documentación incluye ejemplos de uso.

4.1.1. Compilación con la librería UserLibAlMarDuino

Para el uso de la librería **UserLibAlMarDuino** se dispone de ayuda on-line accesible con doble-click sobre el archivo **Client/help.html** de la carpeta de descarga. A partir de dicho enlace, mediante el navegador se puede acceder a toda la documentación para el uso de esta librería desde C/C++.

Los servos se encuentran conectados a los pines PWM 0 a 3 del Arduino simulado (no son pines físicos). La clase Servo de Arduino es funcional para el uso con el simulador (namespace AlmarDuino, ver documentación html).

La aplicación solicitará nombre de aplicación, usuario, clave, etc. , que deben ser los mismos que se pasan a la función **Simulator_ConnectWss()** (ver ejemplo en 4.1.1). Se puede evitar introducir repetidamente estos argumentos estableciéndolos en la línea de comandos (en Qt Creator: Projects (ícono a la izquierda) → Run → Command line arguments):

```
--username="your user name " --passwd="the password you received" --ip="127.0.0.1" --port="8080" --appname="AlMarDrawingRobot"
```

5. Interacción con el simulador desde Matlab/Simulink

En el directorio Client/Library/Matlab se dispone de scripts (funciones y clases) que permiten la interacción con el simulador desde Matlab. En Client/help.html se enlaza con la ayuda para el uso de las funciones y clases Matlab suministradas.

6. Interacción con el simulador desde Python

En el directorio Client/Library/python se dispone de scripts (funciones y clases) que permiten la interacción con el simulador desde Python. En Client/help.html se enlaza con la ayuda para el uso de las funciones y clases Python suministradas.