

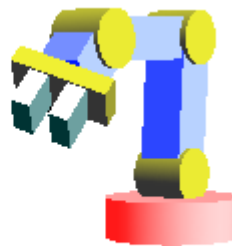
ROBÓTICA

MyOpenLab

Versión 3.0.8.x



Robi2D



Robot Arm 3D

Prof. José Manuel Ruiz Gutiérrez

Índice

1ª Parte. Robot Robi2D

1. Descripción
2. Gobierno del robot
3. Posicionamiento de Imágenes y Gobierno de la Pinza del Robot
- 4.- Movimiento y posicionamiento del Robot
 - 4.1 Control Manual
 - 4.2 Control mediante ratón
 - 4.3 Posicionamiento en un punto
 - 4.4 Seguimiento de posiciones memorizadas en una tabla. Modo Aprendizaje.
5. Ejemplos de manipulación del Robot Robi2D
 - 5.1 Control cíclico de paradas
 - 5.2 Seguimiento de una tabla de posiciones del robot
 - 5.3 Girador
 - 5.4 Entorno básico para la simulación del robot
 - 5.5 Manejo de la Pinza del Robot
 - 5.6 Robot salva-obstáculos

2ª Parte. Robot Arm 3D

1. Descripción General
2. Movimiento y posicionamiento del robot

1ª Parte Robot Robi2D



1. Descripción General:

En esta 1ª parte vamos a explicar las características del elemento de librería de MyOpenLab llamado Robi2D.

El objetivo de este elemento es facilitar al usuario un primer contacto con el mundo de la Robótica en dos dimensiones. Robi2D es un robot virtual en dos dimensiones que es capaz de evolucionar dentro de un “Área Gráfica Canvas” interaccionando con la geometría de su campo de actuación en la que cabe colocar objetos gráficos con los que crear un sencillo escenario de simulación.

El valor pedagógico de este robot virtual es muy grande. Si se trata de utilizarlo en la enseñanza, su utilización y manipulación permitirá al alumno afianzarse en los conocimientos básicos de la robótica y ser capaz de entender la forma de programarlo y ponerlo a realizar tareas...

Robot Canvas (Robi 2D) V. 1.1

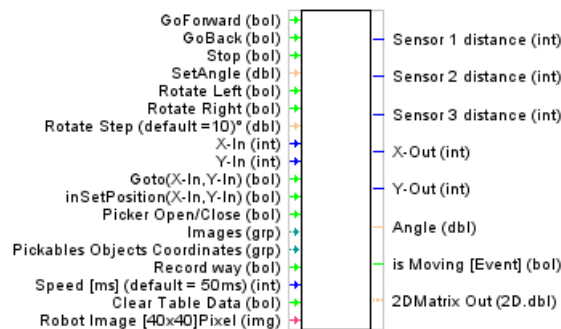


Figura 1

Descripción de señales de E/S del Robot

A la hora de realizar aplicaciones con este elemento son muy numerosas las posibilidades que ofrece en cuanto al control de este en su área canvas de trabajo. Para empezar describimos cada una de las E/S

Nombre	Var.	(E/S)	Descripción
GoForward	bol	E	Mueve el robot hacia delante
GoBack	bol	E	Mueve el robot hacia atrás
Stop	bol	E	Detiene el robot
SetAngle	dbl	E	Asigna valor de ángulo girado
Rotate Left	bol	E	Girar a la izquierda
Rotate Right	bol	E	Girar a la derecha
Rotate Step (default 10°)	dbl	E	Valor de paso de ángulo girado
X-In	dbl	E	Valor de posición coordenada X
Y-In	dbl	E	Valor de posición coordenada Y
Goto (X-In y-In)	bol	E	Orden para posicionar el robot en la coordenada X,Y presente en las entradas X-In, Y-In
inSetPosition(X-In,Y-In)	bol	E	Orden para posicionar el robot en la posición de inicio
Picker Open/Close	bol	E	Orden para abrir y cerrar la pinza
Images	grp	E	Entrada por la que se añaden las imágenes agrupadas en el área canvas del robot
Pickables Objects Coordinates	grp	E	Coordenadas agrupadas de los objetos que el robot puede identificar y coger
Record way	bol	E	Orden que habilita la grabación de las posiciones que adquiere el robot en su movimiento
Speed (ms) Default=50ms	int	E	Velocidad con la que se mueve el robot
Clear Table Data	bol	E	Borra los datos de posicionase almacenados en la tabla.
Robot Image (40x40) pixel	img	E	Imagen con la que se representará el robot
Sensor 1 Distance	int	S	Valor que indica la distancia del robot a un obstáculo medida desde el sensor 1
Sensor 2 Distance	int	S	Valor que indica la distancia del robot a un obstáculo medida desde el sensor 2
Sensor 3 Distance	int	S	Valor que indica la distancia del robot a un obstáculo medida desde el sensor 3
X-Out	int	S	Indica el valor de la coordenada X en donde se encuentra el robot
Y-Out	int	S	Indica el valor de la coordenada Y en donde se encuentra el robot
Angle	dbl	S	Indica el valor del ángulo de giro del robot .
Is Moving [Event]	bol	E	Indica si el robot esta parado o en movimiento
2D Matriz Out	2D.dbl	S	Tabla en la que se recogen las posiciones del robot

El robot se mueve dentro de un área que es modificable en tamaño y también en el aspecto del grid y del fondo.

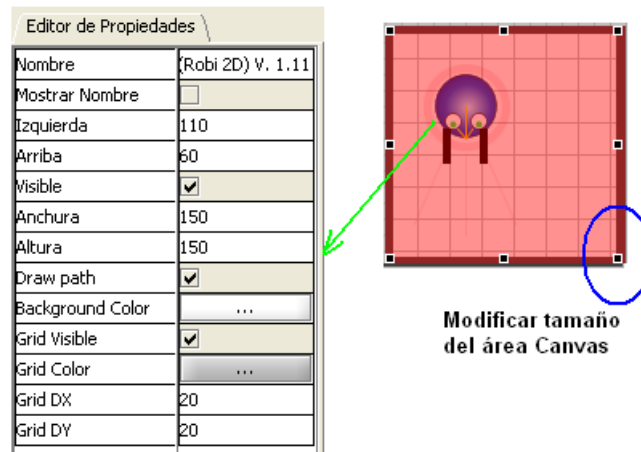
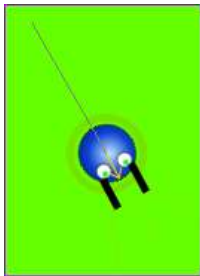


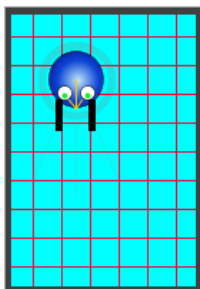
Figura 2

Las características más relevantes son:



Draw path: Cuando se habilita el robot va marcando con una línea su trayectoria sobre el área canvas.

Figura 3



Background Color: Permite modificar el color del fondo
Grid Visible: Permite mostrar el grid en el área canvas
Grid DX: Permite modificar el valor X del cuadro de grid
Grid DY: Permite modificar el valor Y del cuadro de grid
Grid Color: Permite modificar el color del grid.
Visible: Muestra/oculta el área canvas

Figura 4

2. Gobierno del Robot

La idea básica de este elemento es poder realizar con el simulaciones de posicionamiento en el plano XY del Robot pudiendo realizar algoritmos de control que alimenten las señales de entrada del elemento teniendo en cuenta las variables de salida y los parámetros o variables de entrada externa.. En la figura 5 vemos el esquema conceptual de esta idea.

La elaboración de simulaciones con el Robot permite la utilización de las señales que este recibe y devuelve.

Otra manera de interaccionar con el Robot es situando el cursor del ratón dentro del área canvas del Robot y haciendo clic en el botón izquierdo. El Robot se desplazará al lugar

(coordenadas) en donde se ha realizado el clic con el ratón. De esta forma el robot puede seguir trayectorias que le vamos marcando con el ratón.

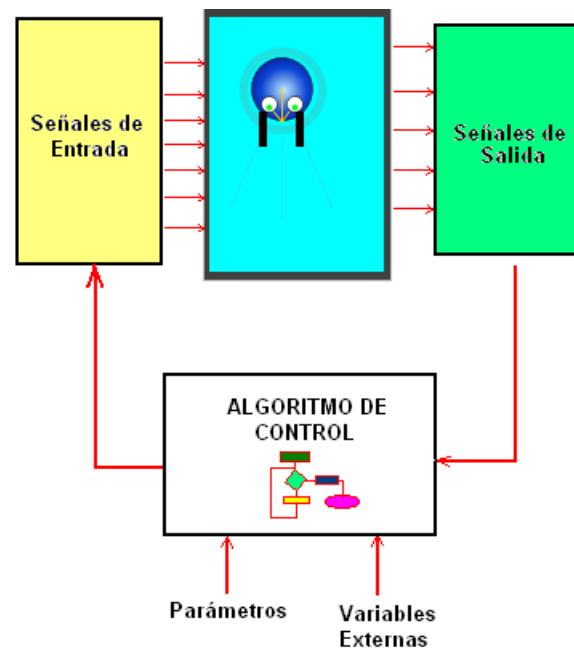


Figura 5

El modelo de control que se muestra es un modelo con realimentación (feed back) lo cual nos permite un elevado control sobre el Robot.

Las señales de salida del Robot nos indican: La posición actual, el estado en el que se encuentra (parado/marcha) y los valores de los sensores que nos indican la distancia a la que se encuentran de un objeto o de las paredes del contorno del área canvas.

Sensores de distancia

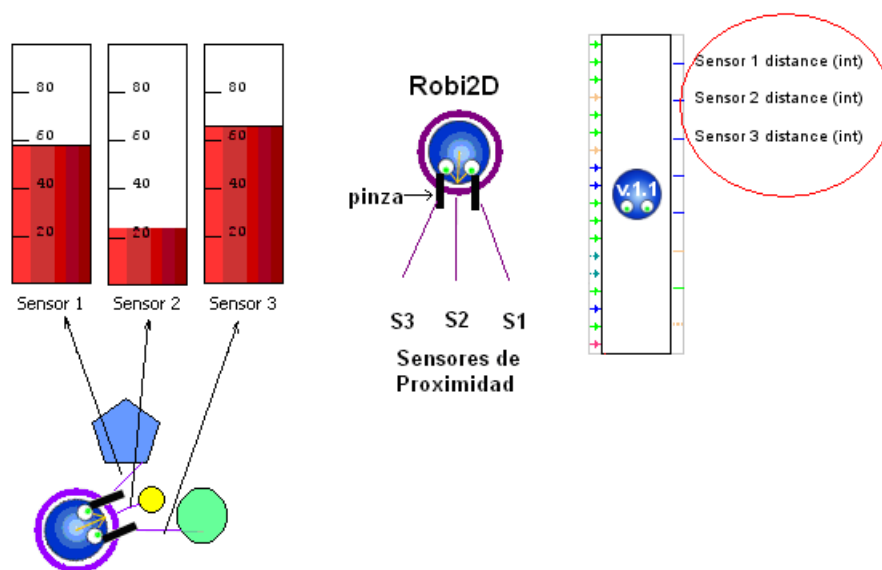


Figura 6

En la figura 6 observamos como los sensores S1, S2, S3 detectan los objetos y envían en sus salidas los valores de las distancias d1, d2 y d3 a la que se encuentran estos objetos.

Selección de imagen de robot

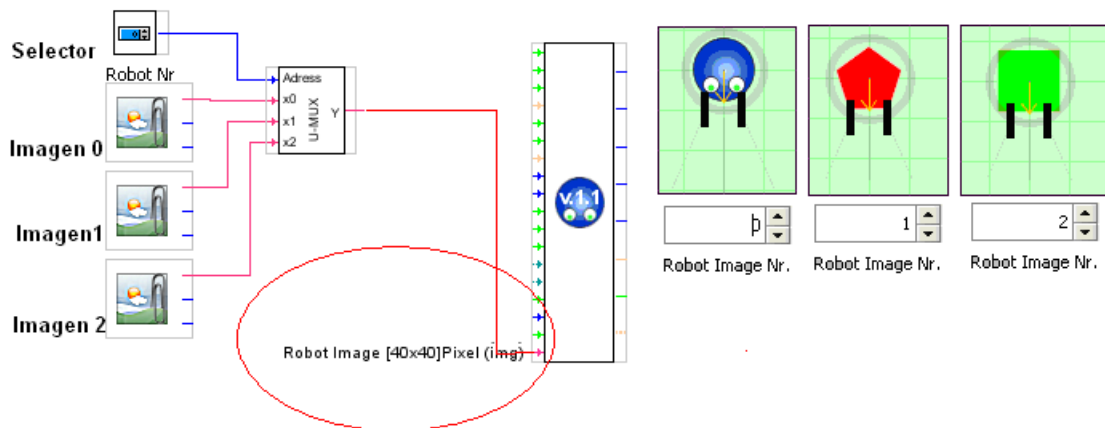


Figura 7

Vemos que en función de la imagen que seleccionemos mediante el Multiplexor gobernado por el selector de número de imagen así entrara por la entrada “Robot Image” una u otra de las tres imágenes que vemos en la figura 7.

3. Posicionamiento de Imágenes y Gobierno de la Pinza del Robot

El Robot puede encontrarse dentro de su campo de trabajo (área canvas) dos tipos de objetos: imágenes normales e imágenes que puede recoger (pequeños círculos negros).

Las imágenes normales solo interactúan en el espacio canvas siendo detectadas por los sensores del robot y cuando este choca contra ellas se detiene y se podrá de nuevo mover el robot si lo giramos y maniobramos para evitar el obstáculo.

Las imágenes se introducen en el área mediante las entradas correspondientes y se definen mediante el fichero que las contenga (gif, jpg, png) y las coordenadas en donde se debe posicionar esta dentro del área canvas. Para introducir las imágenes recurrimos

al bloque de función “Agrupar” .

Las imágenes denominadas “Pickables” son aquellas que el robot puede coger con sus pinzas. Estas solo son círculos negros de tamaño fijo dado que deben estar a la escala correspondiente para poder ser recogidas. Estas imágenes se identifican a través de sus coordenadas X, Y que se introducen mediante el mencionado bloque “Agrupar”. En la figura 8 se ve un ejemplo en el que se han colocados una imagen de cada tipo.

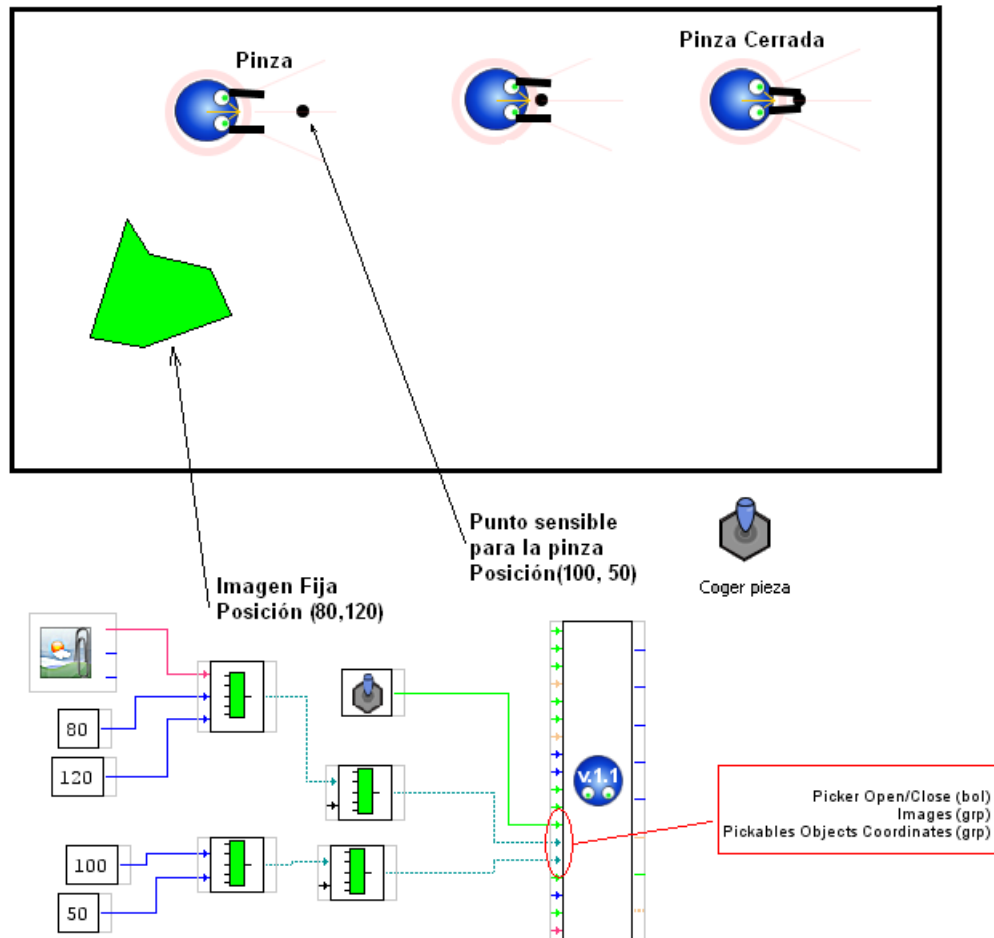


Figura 8

Así mismo se ha colocado un interruptor para el gobierno de la pinza se puede observar en cada una de las tres posiciones del robot su evolución hasta recoger el objeto “pickable”.

Téngase en cuenta que para que la entrada de manejo de pinza sea efectiva (actúe) el robot debe estar parado y lógicamente el objeto a recoger en el área precisa de la pinza.

4. Movimiento y posicionamiento del robot.

De manera básica el movimiento del robot se realiza:

- Para que el robot avance hacia delante o retroceda se hace uso de las entradas de tipo Booleano *GoForward* (avanza) y *GoBack* (retrocede).
- Para que gire se hace uso de las señales: *Rotate Left* (gira izquierda) *Rotate Right* (gira derecha)

El movimiento del robot se puede llevar acabo de las siguientes formas:

1. *Control manual*
2. *Control mediante el ratón*
3. *Posicionado en un punto.*
4. *Seguimiento de posiciones memorizadas en una tabla.*

4.1. Control Manual:

En este tipo de control el robot se mueve con los controles de tipo booleano:

- *Adelante* (GoForward)
- *Atrás* (GoBack)
- *Parar* (Stop)
- *Girar Izquierda* (Rotate Left)
- *Girar Derecha* (Rotate Right)

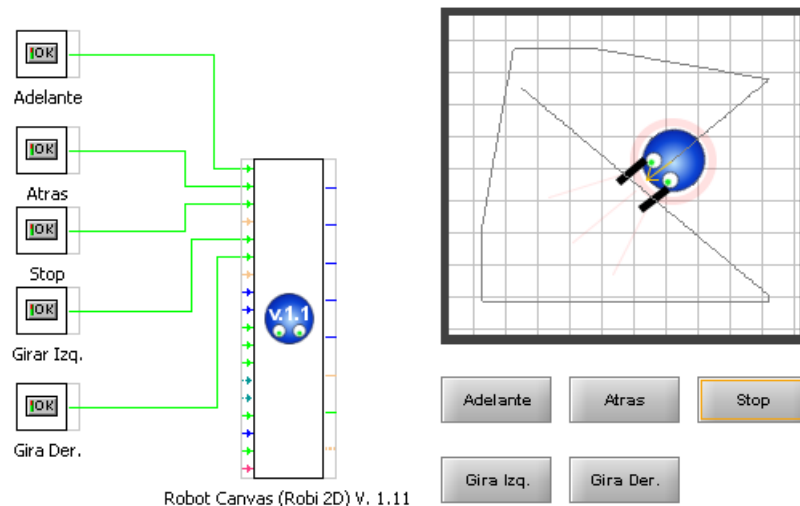


Figura 9

- Tengamos siempre en cuenta que el avance, retroceso y parada del robot se ejecutan con un simple pulsación en el la entrada correspondiente. En este caso el Robot si por ejemplo se le ordena avanzar se pulsa sobre el pulsador “Avanzar” (una pulsación) y el robot inicia su marcha hacia delante hasta que se encuentra un obstáculo y se detiene. Si el obstáculo se quita o el robot se gira continua con su avance. Esto significa que las tres órdenes que hemos mencionado cuando se dan actúan por impulso y además anulan la anterior.
- Las ordenes que hacen girar al Robot son dos “Gira Izquierda” y “Gira Derecha” que al contrario de las anteriores se den mantener activadas (valor 1) para el que el robot gire de manera continua y si damos un impulso gira un ángulo (el valor de este ángulo se puede modificar mediante la seña de entrada “Rotate Steep” que por defecto son 10°)

El robot se moverá a una velocidad que por defecto esta colocada a un valor de retardo de 50 ms. Pero que podemos modificar simplemente variando el valor de la señal de entrada de tipo entero “Speed”. Cuanto mayor hacemos el valor de esta señal mas lento se mueve el Robot la mayor velocidad se alcanza cuando esta variable es igual a cero, pero este valor no conviene ponerlo.

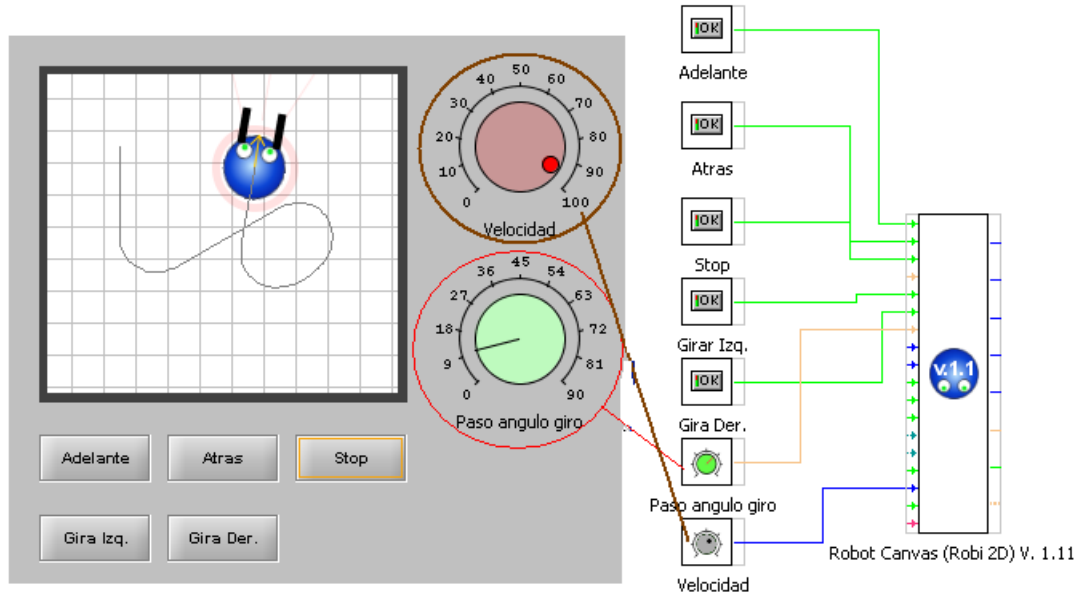


Figura 10

4.2. Control mediante el ratón.

Se puede mover el ratón dentro de área canvas pulsando simplemente con el ratón en un punto. El robot se dirigirá a ese punto y se detendrá una vez que lo alcance.

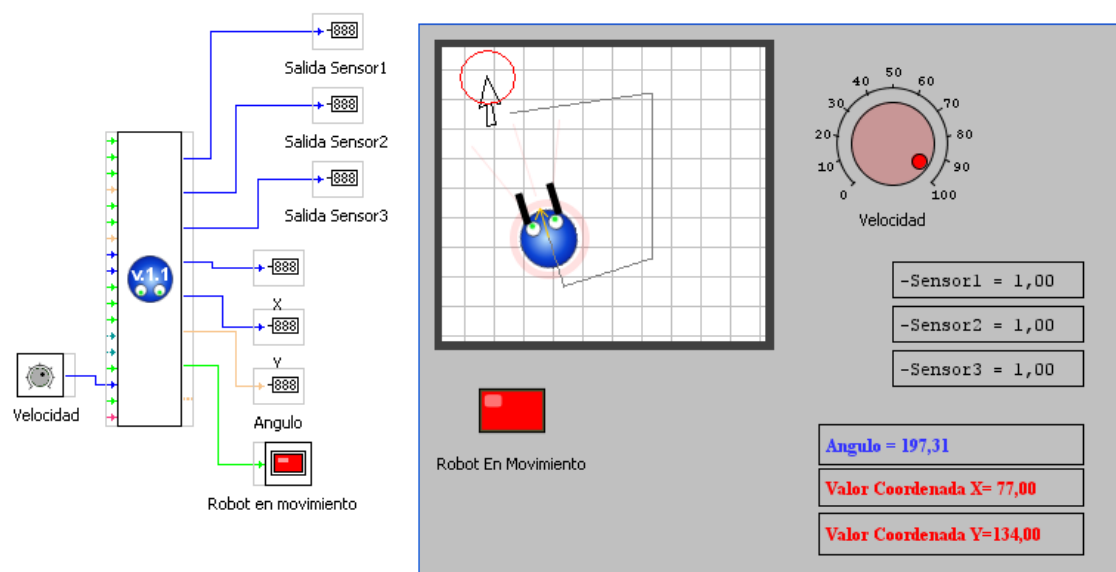


Figura 11

En la figura 11 se han puesto los indicadores numéricos de las señales de salida del robot que indican su posición XY, su Angulo Girado y los valores de los Sensores 1,2,3. También se ha colocado el indicador de Robot en Movimiento para saber cuando esta parado o en marcha el robot.

Las señales utilizadas en este tipo de control del Robot son:

• <i>Sensor Distancia 1</i> (Sensor 1 Distance)	int
• <i>Sensor Distancia 2</i> (Sensor 2 Distance)	int
• <i>Sensor Distancia 3</i> (Sensor 3 Distance)	int
• <i>Salida Coordenada X</i> (X-Out)	int
• <i>Salida Coordenada Y</i> (Y-Out)	int
• <i>Angulo</i> (Angle)	dbl
• <i>Movimiento Robot</i> (Is)Moving [Event]	bol

4.3. Posicionado en un punto.

Puede interesarnos que el robot se posicione en un punto concreto o vaya a un punto concreto, cuyas coordenadas queramos darle exactamente.

En este caso se recurre al control mediante posicionamiento por coordenadas XY.

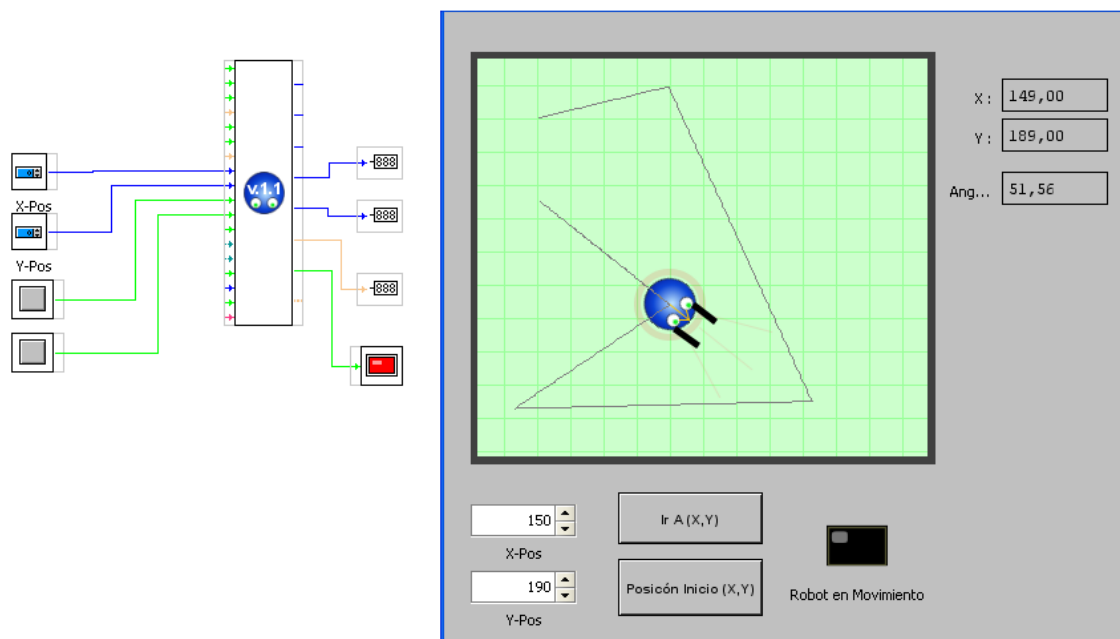


Figura 12

El procedimiento es muy sencillo. Se trata de colocar en las entradas “X-In” e “Y-In” los valores de las coordenadas X e Y en donde queremos llevar al robot...

- Cuando ponemos el valor 1 (impulso) en la entrada “*Goto (X-In, Y-In)*” el robot inicia su marcha y se dirige a esa coordenada.
- La señal *inSetPosition (X-In, Y-In)* (impulso) posiciona de manera instantánea el robot en el lugar que indiquen las coordenadas X e Y sin realizar recorrido alguno sin variar el valor del ángulo de giro que tuviese el robot antes de dar la orden.

4.4. Seguimiento de posiciones memorizadas en una tabla. Modo Aprendizaje.

Esta forma de realizar desplazamientos del robot en su área canvas es muy útil siempre que queramos emular el modo de aprendizaje con nuestro robot.

Si tenemos habilitada la entrada (en valor 1) “Record way “ en ese caso todos los movimientos del robot se grabarán en una tabla mientras que mantengamos la señal activada y siempre que el robot se mueva.

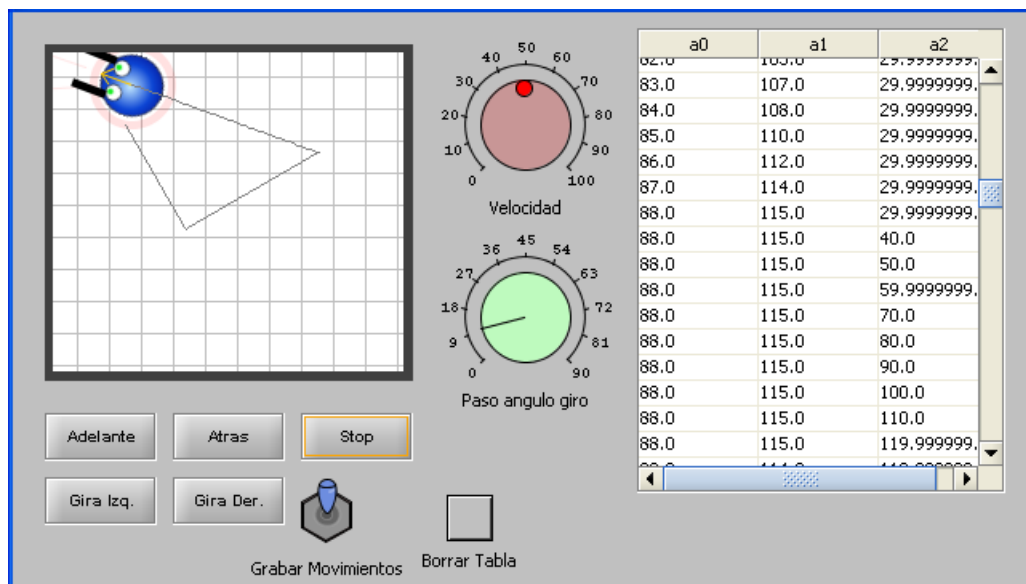


Figura 13

Las señales que se utilizan de manera especial en este modo de control del robot son:

- *Grabar trayectoria* (Record way) bol
- *Borrar Tabla* (Clear Table Data) bol
- *Matriz datos posiciones* (2D Matriz Out) 2D.dbf

5. Ejemplos de manipulación del Robot Robi2D

A continuación vamos a describir algunos ejemplos básicos para familiarizarse con el manejo del robot y conocer algunas de sus posibilidades.

5.1 Entorno básico para simulación del Robot

En el siguiente ejemplo se han dispuesto todos los controles del elemento Robot con el fin de poder experimentar con cada uno de ellos.

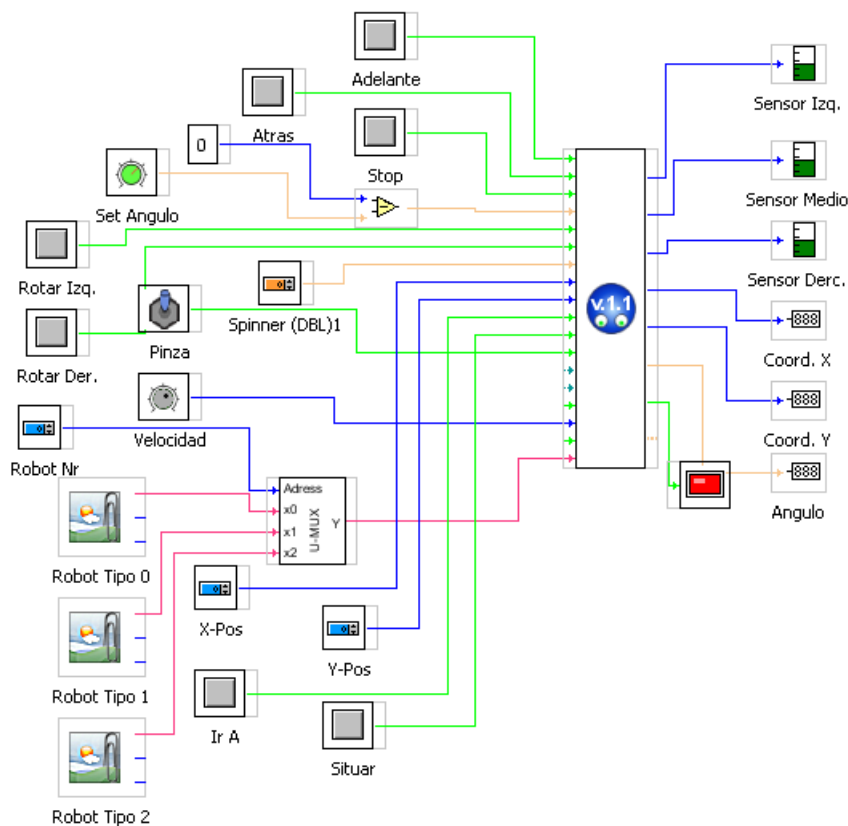


Figura 14

Se sugiere montar esta aplicación y manipular los controles con el fin de poder descubrir las múltiples posibilidades que tiene Robi2D

Nótese también que se pueden modificar los tipos de robot seleccionado en el correspondiente control **Robot Nr->Selección Tipo de Robot**

En la siguiente figura 15 podemos identificar el panel lógico en donde figura el cableado de la aplicación con todos los elementos de entrada y salida que luego aparecerán en el panel frontal.

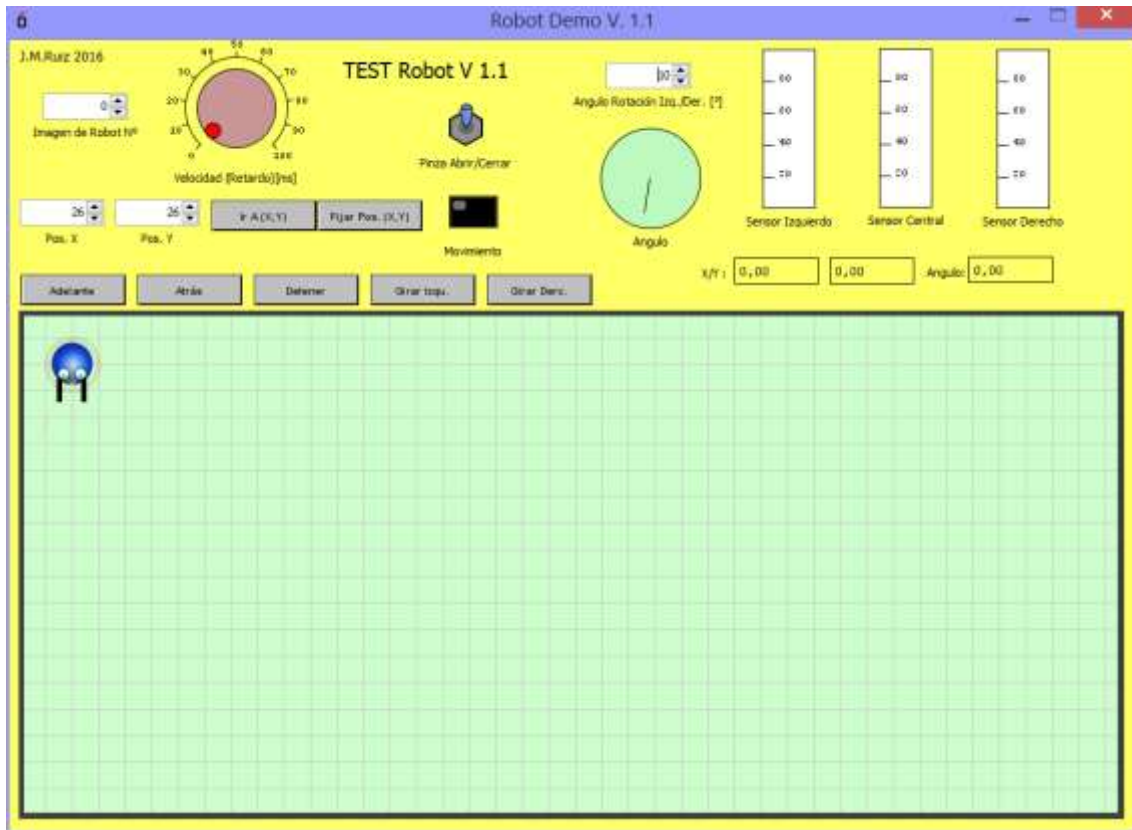


Figura 15

5.2 Control cíclico con paradas.

El objetivo de esta aplicación es conseguir que el robot se desplace en sentido horizontal en un movimiento de ida y vuelta cíclico y que cada vez que pase frente al la señal de Stop se tenga un tiempo. En el semáforo se pondrá la luz roja cuando este detenido y verde cuando se reinicie la marcha. Cuando llega el robot a los extremos izquierdo o derecho de su recorrido se detiene y esta un tiempo parado (semáforo en rojo).

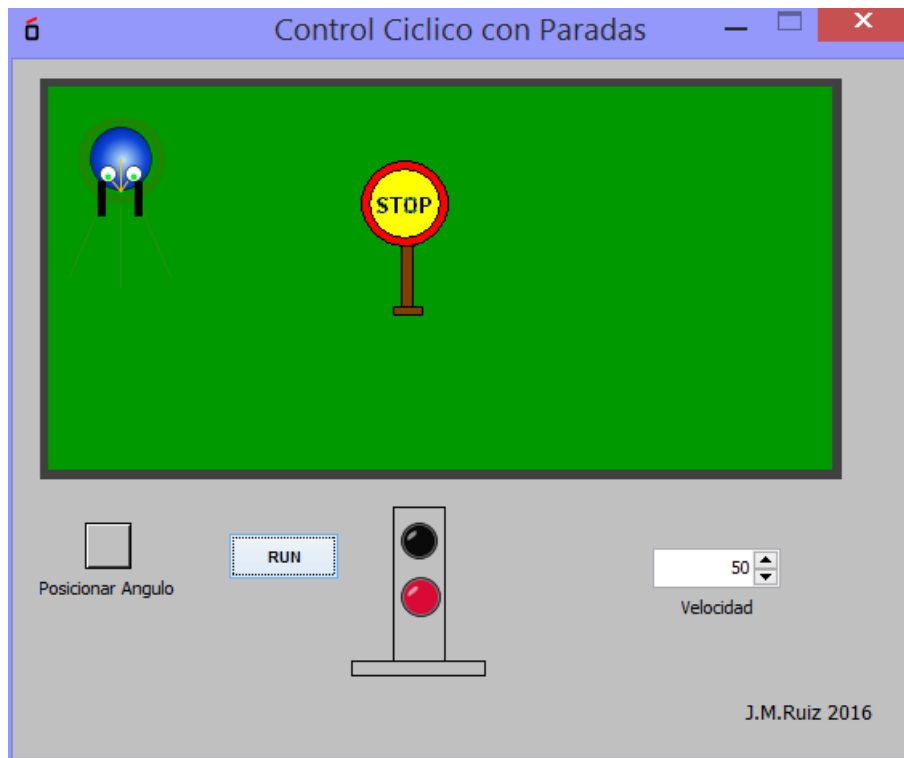
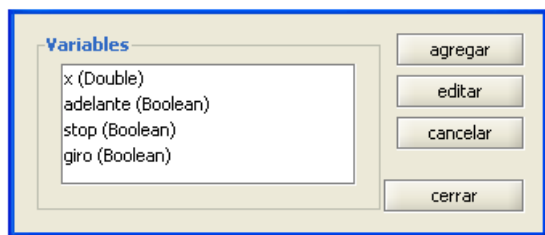


Figura 16

En la figura 16 vemos el aspecto de la pantalla en modo simulación. El botón RUN sirve para iniciar el proceso.




Lo perímetro que debemos hacer es definir unas señales que interaccionaran con el elemento Robot y que también lo harán con el diagrama de flujo que controle la aplicación.

En la figura 17 vemos el aspecto de las señales que gobiernan el Robot:

La variable “**x**” representa el espacio recorrido por el robot (Coordenada X de su posición).
 La variable “**adelante**” es la que ordena al robot que inicie su marcha hacia adelante.
 La variable “**stop**” detiene el robot
 La Variable “**giro**” hace que el robot gire el ángulo que

Figura 17

previamente le tenemos puesto en la entrada “Set Angle” del Robot.

Las variables definidas se entregan o recogen haciendo uso de los elementos  .

La imagen de la señal de Stop se introduce en el elemento Robot a través de la entrada “Images”

Los parámetros son los siguientes:

Set Angle=90°
 Rotate Step=180°
 X-In=50
 Y-In=200

Cuando se inicia la simulación el Robot se deberá situar automáticamente (ya que la señal InSetPosition(X-In, Y-In)=1) en las coordenadas (50,100) y girado 90°

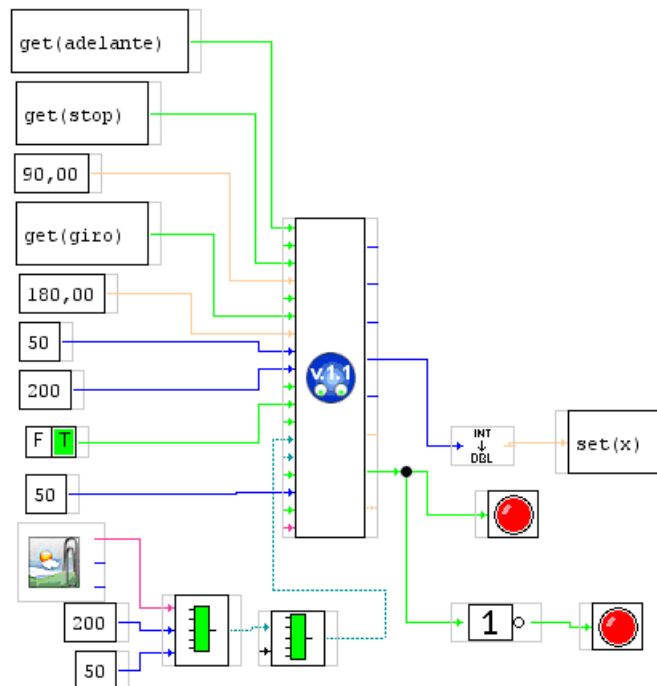


Figura 18

En la figura 18 vemos el aspecto del elemento Robot con sus señales de entrada y de salida.

En la figura 19 vemos el organigrama de control que realiza el movimiento que hemos propuesto que realice.

Como datos significativos en el movimiento del robot tenemos los siguientes:

- Detección del robot frente al Stop cuando marcha hacia la derecha $x > 200$
- Detención cuando llega al extremo derecho $x > 400$
- Detención del robot frente al Stop cuando marcha hacia la izquierda $x < 240$
- Detención en el extremo izquierdo $x < 80$
- Tiempos de detención de robot 2000 ms

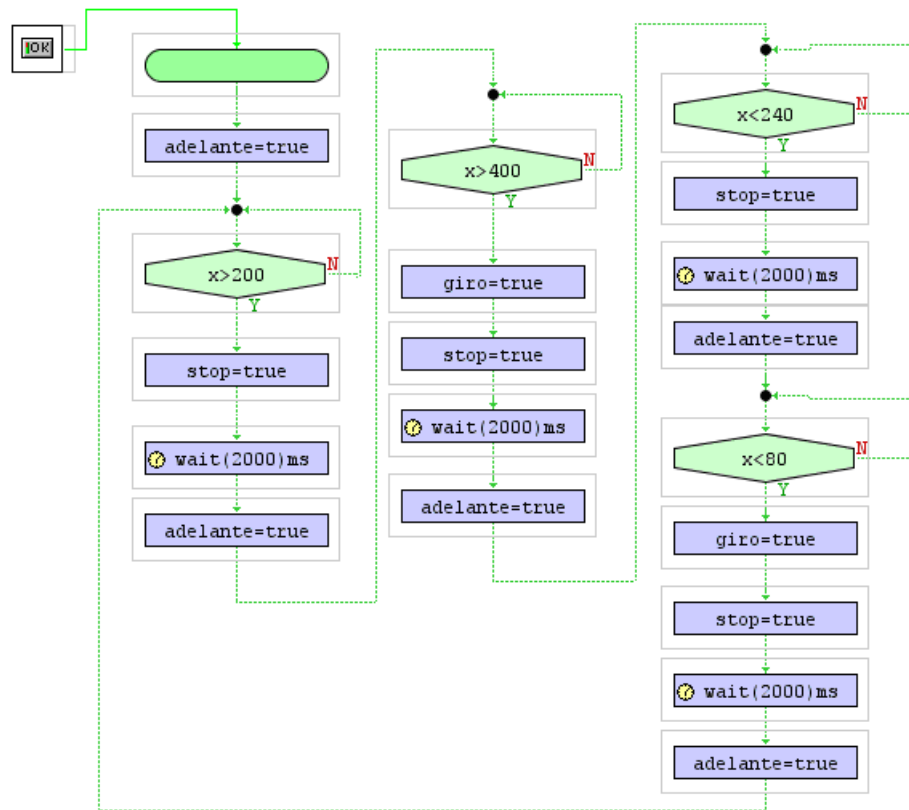
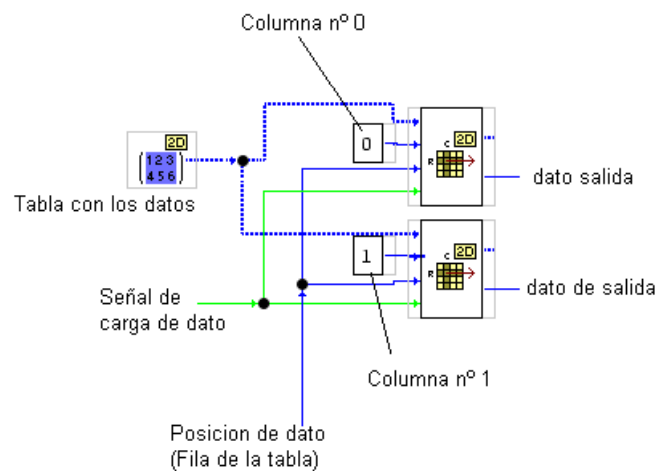


Figura 19

5.3 Seguimiento de una tabla de posiciones del Robot

El siguiente ejemplo muestra como se pueden leer de una tabla las posiciones X e Y que el robot ira recorriendo a lo largo del área cava.



En una tabla se escriben los datos y después se van leyendo y poniendo en las entradas X-In e Y-In del robot dando las correspondientes ordenes para que este vaya al lugar indicado. (señal *Goto (X-In y-In)*)

Este ejemplo permite desarrollar otros más complejos en los que asignemos al robot recorridos más complejos e incluso la posibilidad de recoger objetos.

Figura 20

Para incluir más datos en la tabla

bastará con añadirle columnas y luego, tal como se ve en el esquema del panel lógico se separan cada una de las columnas de datos y se llevan a la entrada correspondiente.

En la figura 21 vemos el esquema funcional del ejercicio.

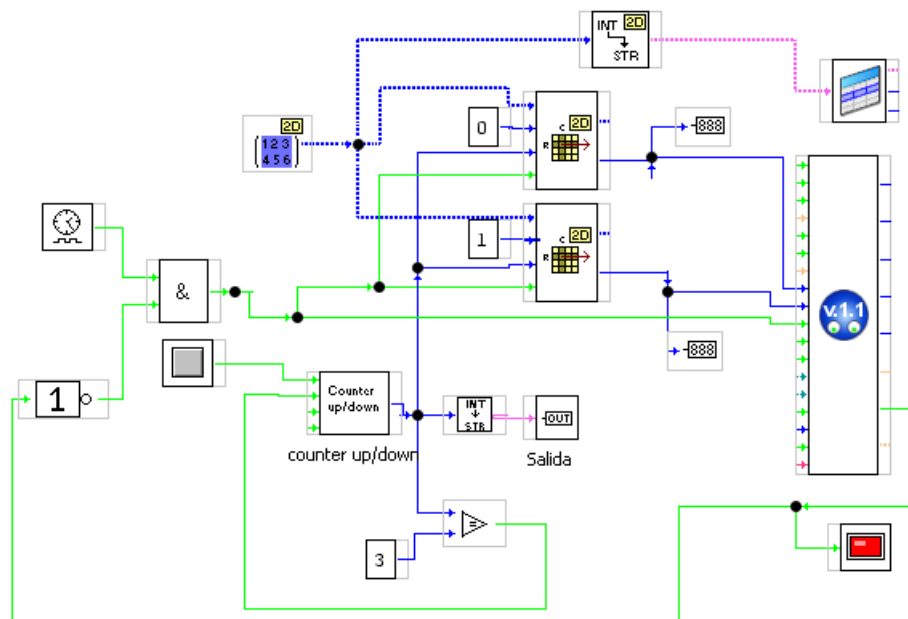


Figura 21

En la figura 22 vemos el panel frontal en modo simulación.

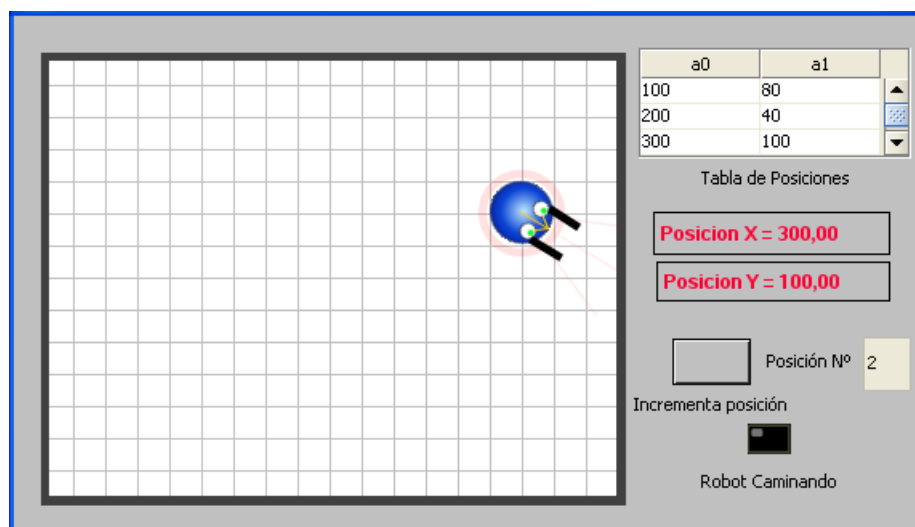


Figura 22

5.4 Girador

Se trata en este ejemplo de controlar el movimiento del robot haciendo hincapié en el comando de giro.

Básicamente lo que se persigue es que mediante una orden el robot se ponga a girar de manera continua, pudiendo hacerlo a izquierdas o a derechas dependiendo de cómo se te actúe.

Las variables del modelo son las siguientes:

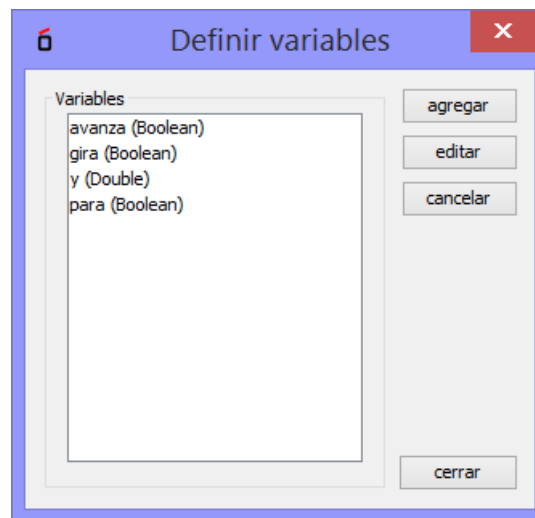


Figura 23

La variable **gira** es la que manda girar al robot.

La variable **para** detiene el giro del robot

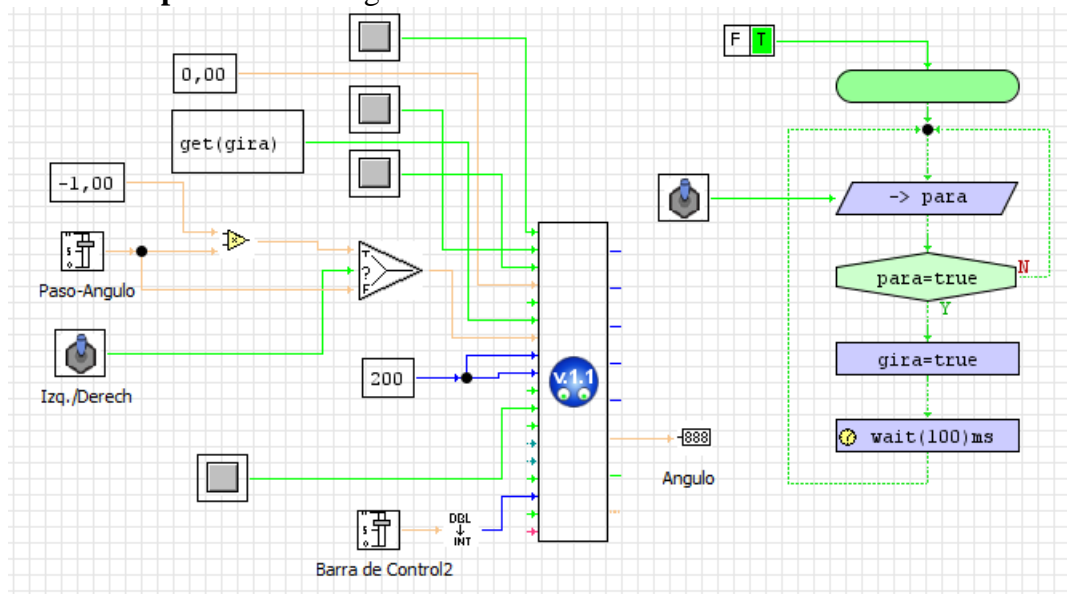


Figura 24

Los parámetros y resto de variables que controlamos en el ejemplo son:

- *Velocidad de desplazamiento* del robot (controla la velocidad de avance/retroceso del robot)
- *Angulo girado* (indica el ángulo que gira en cada paso)
- *Giro a derecha/izquierda*.
- *Parar* (detiene el movimiento de avance o retroceso del robot)
- *Avanzar* (ordena el avance del robot)
- *Retroceder* (ordena el retroceso del robot)

La generación de los impulsos de giro se realiza mediante el organigrama , que como vemos, siempre que la variable **para=1 (true)**, con intervalos de **200 ms** activa la señal gira

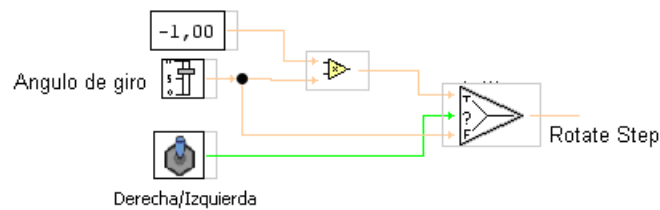


Figura 25

Para conseguir que los giros sean a la izquierda o derecha lo que hacemos es cambiar el valor del ángulo a girar de tal manera que cuando el ángulo es negativo el giro es para la derecha y cuando es positivo el giro es para la izquierda

La posición de inicio se ha designado en la coordenada (200,200), de tal manera que cuando se actúa sobre el pulsador “**Inicio**” el robot se coloca instantáneamente en esa posición.

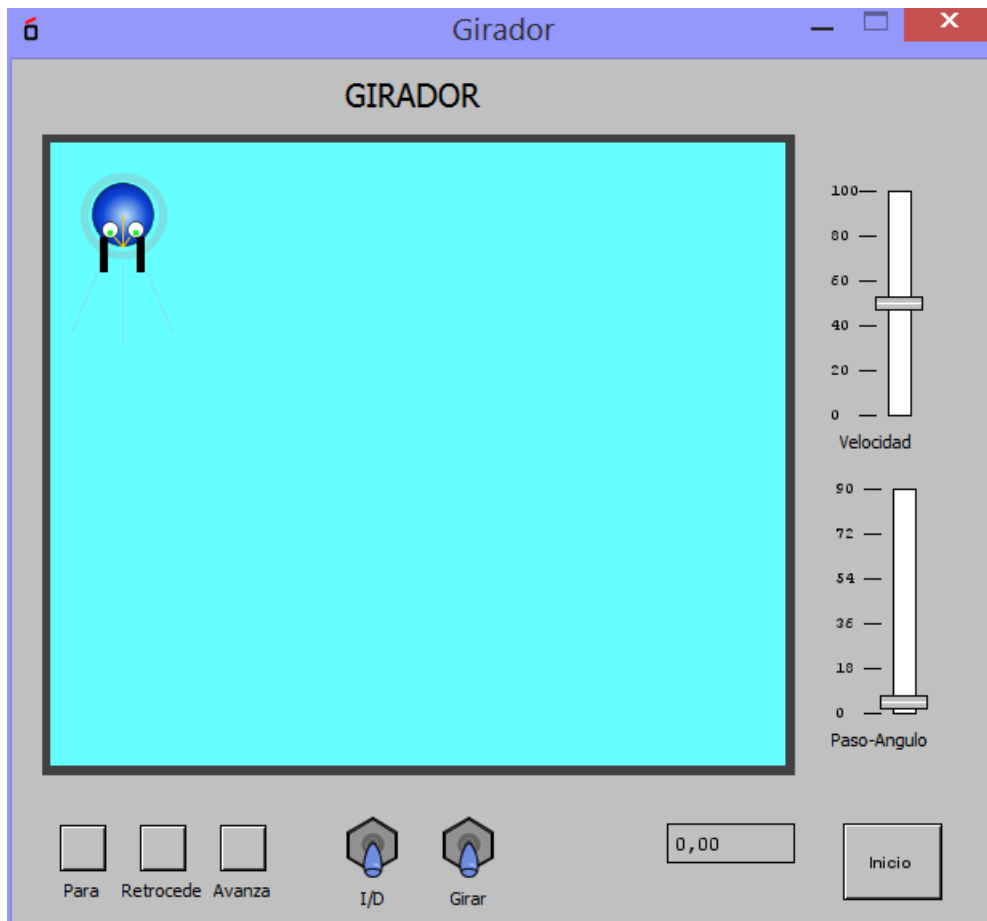


Figura 26

La figura 26 muestra el aspecto del panel frontal en modo simulación.

5.5 Manejo de la Pinza del Robot

En este ejemplo se trata de realizar ensayos de recoger un objeto con la pinza del robot y transportarlo a otro lugar del área canvas.

A continuación vemos el esquema del panel lógico con los elementos conectados al robot, así mismo se ve que el conjunto de las imágenes se ha encapsulado en un submodelo que también vemos en la figura 27.

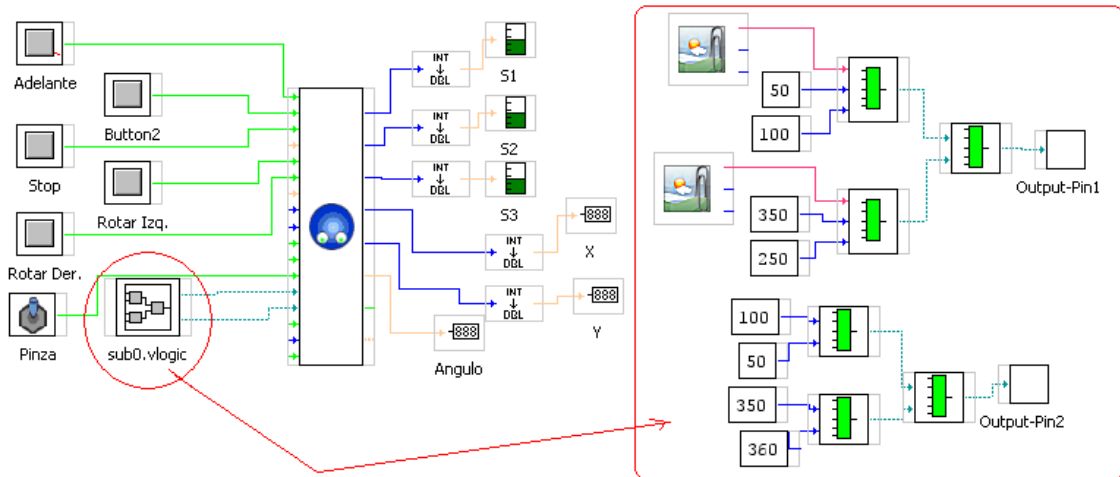


Figura 27

El panel frontal en modo simulación quedaría como la siguiente figura

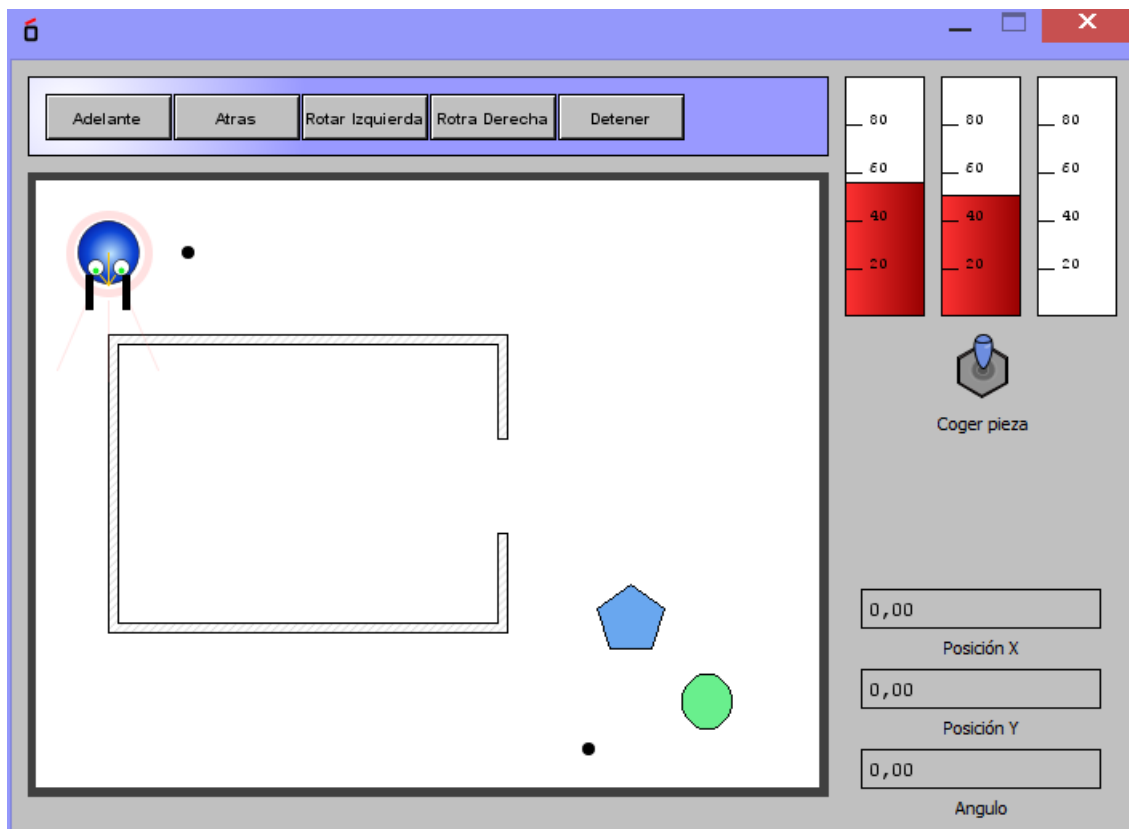


Figura 28

5.7 Robot salva-obstáculos

Se trata de crear una aplicación que permita que el robot se mueva en el área canvas y que cuando llegue a chocar con una de las paredes gire hasta que consiga seguir su marcha.

La idea principal está basada en detectar el valor de uno de sus sensores de distancia (en nuestro caso utilizaremos S1 –sensor izquierdo-) y cuando los valores de la distancia

estén comprendidos entre 0 y 50 se active una señal que llamaremos **choque** la cual activara la ejecución de una rutina (diseñada en el organigrama) que haga girar al robot un valor que esta recogido en la variable **ángulo**.

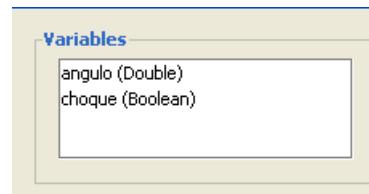


Figura 29

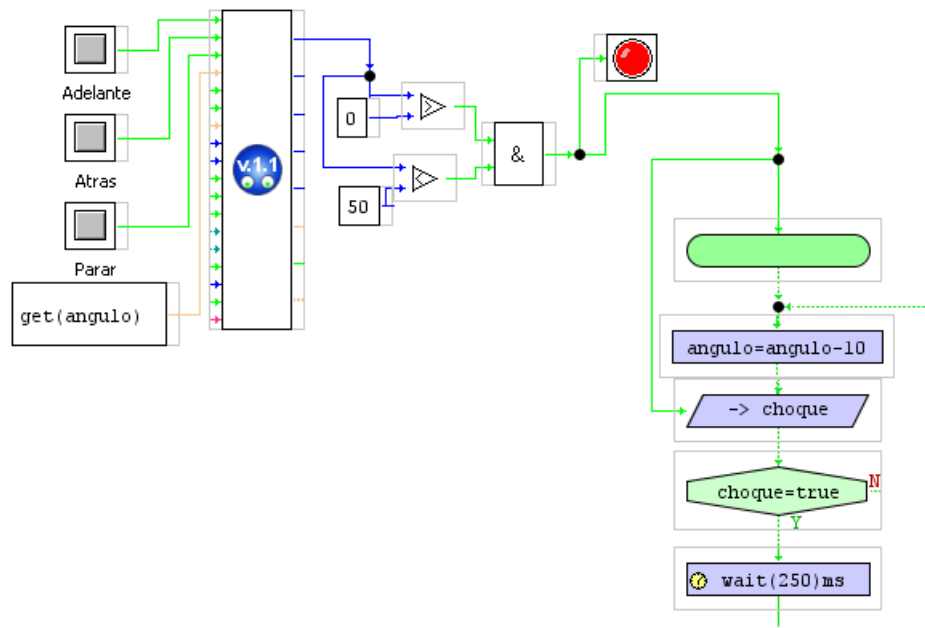


Figura 30

Los impulso que hacen girara el robot se producen cada 250 ms.

Una vez que el robot supera la pared haciendo los giros necesarios se desactiva la señal choque y continúa avanzando.

No olvidemos que los sensores van en la parte delantera del robot, eso significa que si mandamos mover para atrás al robot y choca no se ejecutará el giro y el robot quedará detenido.

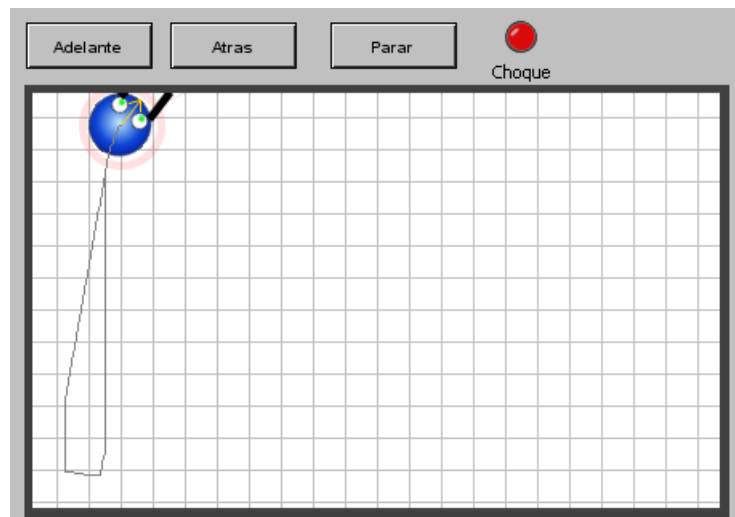
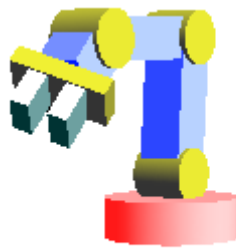


Figura 31

En la figura 31 vemos el aspecto del panel frontal en modo simulación.

2ª Parte Robot Arm 3D



Robot Arm 3D

1. Descripción General:

El Robot Arm 3D es un componente para la simulación de un brazo robot en tres dimensiones que permite comprender la cinemática de un robot de una manera sencilla.

Vemos en la figura 32 la ventana de componente en la que se muestran las señales de entradas del robot. También se pueden apreciar los grados de libertad o ejes de movimiento del brazo que son tres (Z1,Z2,Z3) más la Muñeca (Hand) y la Base (Basis)

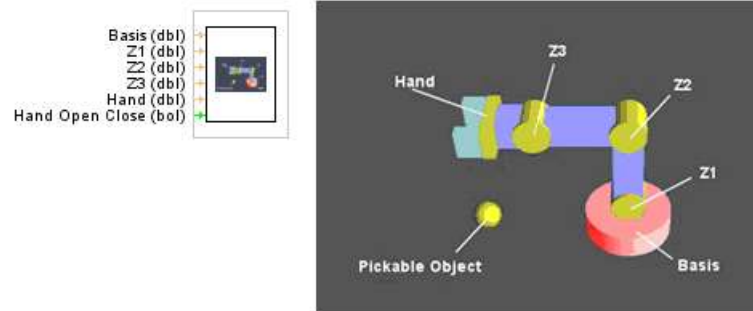


Figura 32

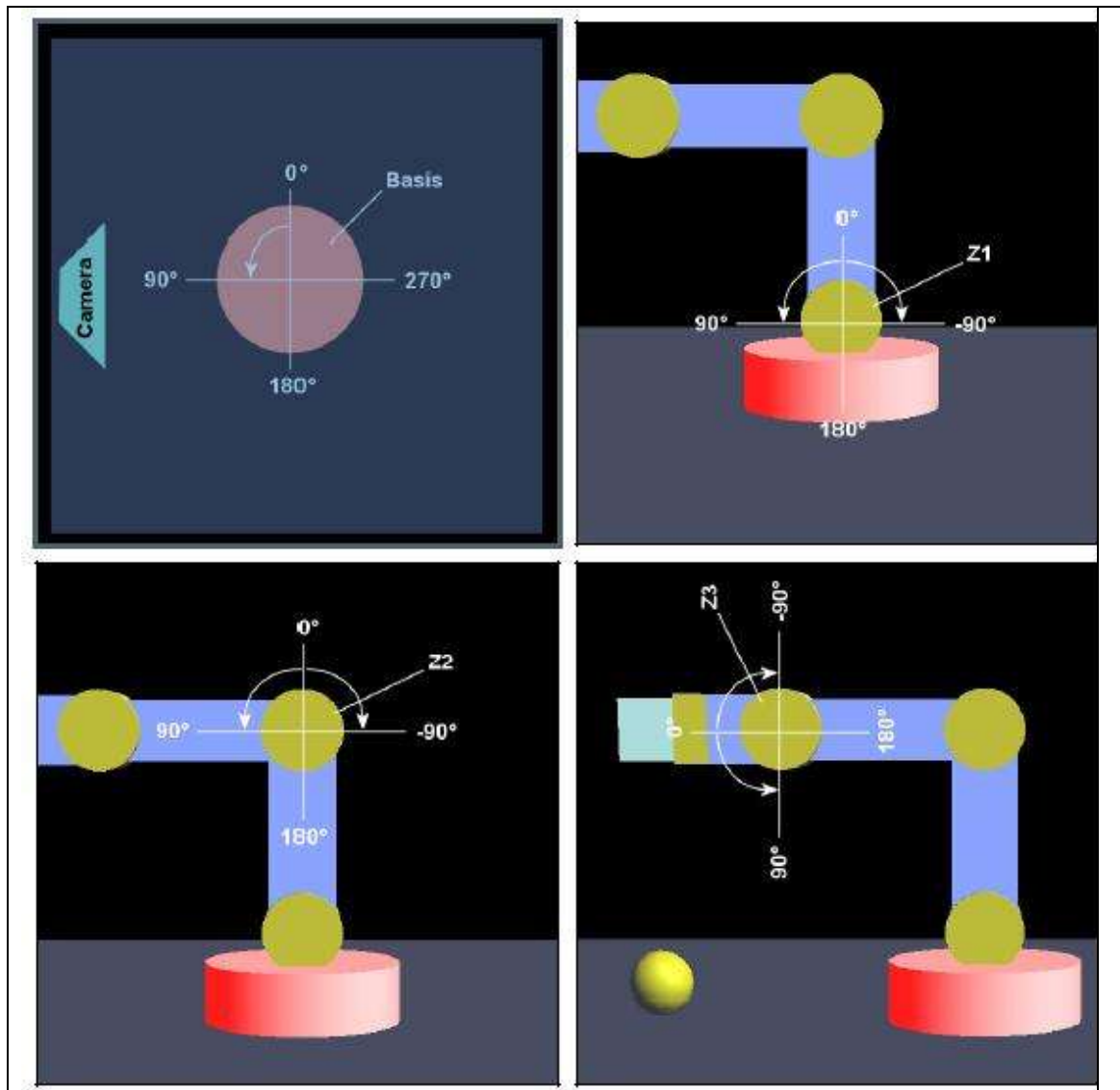
En la muñeca el Robot tiene una pinza mediante la cual es capaz de recoger un objeto (esfera), para lo cual se utiliza la señal de entrada “Hand Open Close” (abrir y cerrar pinza)

En la tabla que se muestra a continuación quedan descritas las señales de gobierno del Robot Arm 3D

Nombre	Var.	(E/S)	Descripción
Basis	dbl	E	Mueve la base del robot
Z1	dbl	E	Mueve eje Z1
Z2	dbl	E	Mueve eje Z2

Z3	dbl	E	Mueve eje Z3
Hand	dbl	E	Girar muñeca
Hand Open Close	bol	E	Abre y cierra pinza

Los movimientos del robot en cada uno de sus ejes quedan mostrados en la siguiente figura, en donde



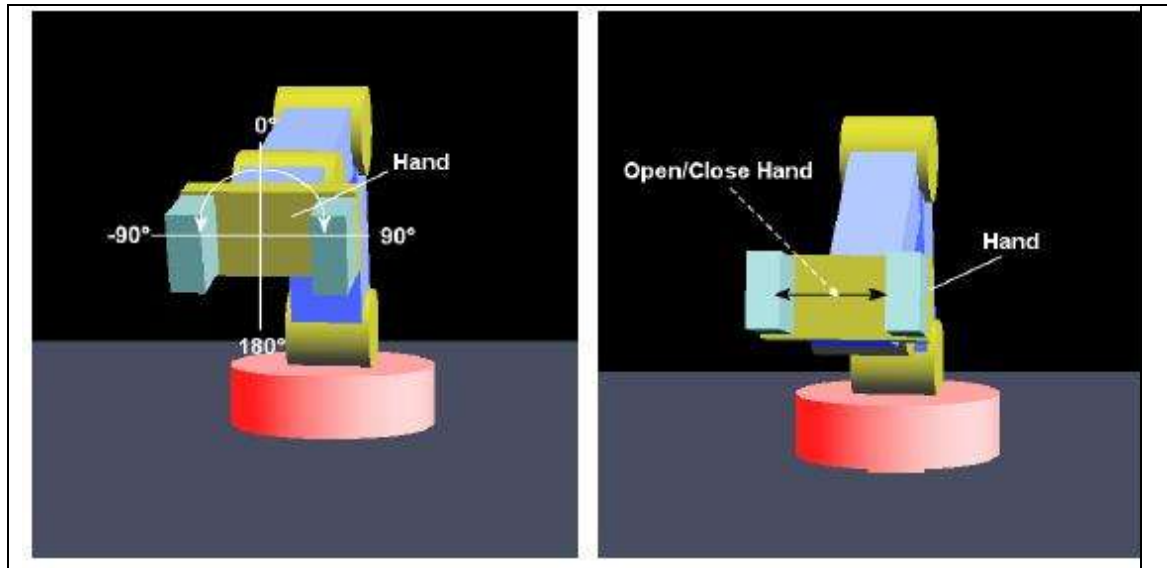


Figura 33

Para coger un objeto con la pinza solo es necesario que la distancia a la que se encuentre este comprendida dentro de un radio de distancia determinado, no es preciso el encaje total del objeto dentro de la pinza.

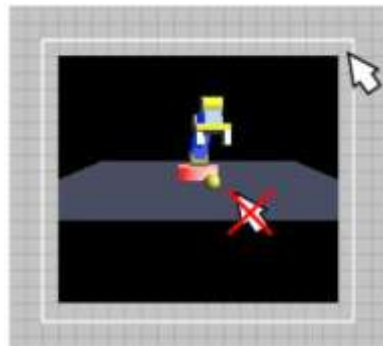


Figura 34

Para poder agrandar o achicar el área canvas del Robot 3D se debe pulsar en el marco del área y aparecerá el contorno sobre el que podremos realizar la ampliación o reducción del área tal como se indica en la figura.

Movimiento de la Cámara

Para mover la cámara de posición es necesario seleccionar primero el Robot (haciendo clic estando sobre el) y luego presionando

Para cambiar la vista

LEFT	Izquierda
RIGHT	Derecha
UP	Arriba

DOWN Abajo

Movimiento de cámara

W Ir adelante (Go Forward)
 E ir hacia atrás (Go Back)
 A Ir a la izquierda (Go Left)
 S Ir a la derecha (Go Right)

La cámara solo se puede mover estando en modo runtime

2. **Movimiento y posicionamiento del robot.**

Para el movimiento de este robot no olvidemos que debemos poner en las entradas correspondientes a los ejes, base o muñeca los valores analógicos equivalentes al valor de ángulo que deseamos girar.

Recordemos que los rangos de valor de ángulo de cada eje son los siguientes:

Eje de Giro	Min. (°)	Max (°)
Basis	0	360
Z1	-90	90
Z2	-130	130
Z3	-120	120
Hand	-90	90

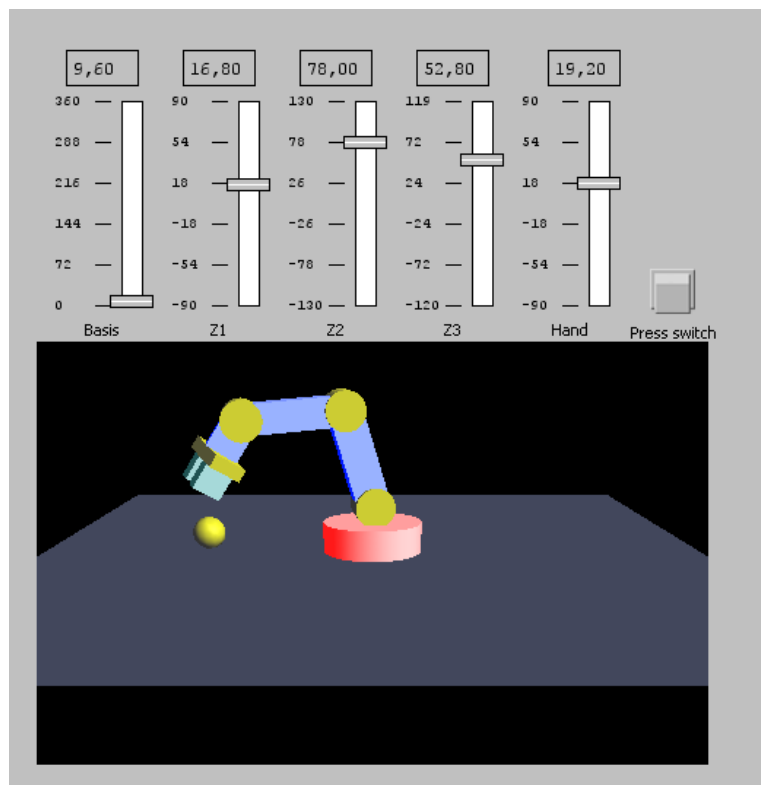


Figura 35