



**Universidad Nacional de Lomas de
Zamora**

Facultad de Ingeniería



INFORME DE PRÁCTICA PROFESIONAL SUPERVISADA

CIM Robotica - FI-UNLZ

**PUESTA A PUNTO Y MEJORA DEL BRAZO ROBÓTICO
SCORBOT EX IX**

INTEGRANTES

- Audisio, Juan Pablo - 43.671.648
- Gomez Franco Gabriel - 41.451.020
- Reyna, Valentin - 43.798.677



Índice

Reservado a la facultad para la evaluación.....	2
LUGAR DE REALIZACIÓN DE PPS	3
Detalle del trabajo realizado.....	4
Marco teórico	4
HARDWARE	4
Robot / controlador	4
Estación de trabajo	5
Círculo eléctrico del controlador.....	5
PROGRAMACIÓN.....	7
Lenguaje.....	7
Vectores	7
Programas	7
Comandos y Recomendaciones	8
Variables	8
Programas y uso de la estación de trabajo.....	9
PROGRAMAS CREADOS.....	9
LOBBY	9
GETXX	9
PUTXX	9
PR1, PR2, PR3 Y PR4	9
CONSIDERACIONES DE USO.....	10
Regulación de velocidad de cierre del gripper	10
Teach	10
Activación de ejes.....	10
Encendido	10
Apagado	11
Conclusiones	12



Reservado a la facultad para la evaluación



LUGAR DE REALIZACIÓN DE PPS

La práctica tomó lugar en la Facultad de Ingeniería de la Universidad de Lomas de Zamora, en el laboratorio CIM de robótica.

Las instalaciones cuentan con una línea de proceso completamente automatizada, suministrada por ESHED Robotics. Esta línea de manufactura incluye una cinta transportadora, un almacén y tres estaciones de trabajo. Tanto el almacén como cada estación de trabajo disponen de un robot.

El robot del almacén se encarga de la selección y colocación de la materia prima indicada en la cinta transportadora. Por su parte, los robots de cada estación de trabajo realizan las operaciones específicas de su estación. Cada robot es controlado por su propio controlador, y todos estos, junto con el controlador de la cinta transportadora, son gestionados por un tablero central “Manager”.



Detalle del trabajo realizado

Se desarrolló código para controlar la estación de trabajo número 2 de la línea de producción del laboratorio CIM. Dicho código se encarga de ejecutar operaciones automatizadas, basándose en señales de entrada y salida, bajo la supervisión del tablero central "Manager".

La línea de producción consta de 4 estaciones de trabajo, en cada una de las cuales se realizan operaciones específicas según programas preestablecidos.

Mediante la lectura de 4 entradas del controlador, el código ejecuta el programa correspondiente. Este proceso incluye retirar una bandeja, realizar una operación sobre ella y devolverla a la cinta de transporte. Al finalizar la operación, se envía un pulso de recibo (ACK) al "Manager".

Marco teórico

HARDWARE

Robot / controlador

El brazo robótico utilizado es el SCORBOT EX IX de 5 grados de libertad interpolables, montado sobre un *Lineal Slidebase* de la marca ESHED ROBOtec que incorpora 1 grado de libertad de translación (no interpolable). Para controlarlo se utiliza el controlador de la marca ESHED ROBOtec, el cual funcionará como intermediario de I/O y conexión con el PC.

El sistema también cuenta con un *Teach Pendant* de la marca ESHED ROBOtec conectado al controlador permitiendo un control a pie de máquina y facilitando la programación y testeo.

Imagen 1.1. Brazo Robótico SCORBOT. Imagen 1.2. Teach Pendant. Imagen 1.3. Lineal Slidebase





Imagen 1.4. Controlador del brazo robótico. Imagen 1.5. Tablero Manager.

Estación de trabajo

El brazo robótico previamente mencionado funciona en conjunto con un *PC manager* central el cual controla la estación. El robot recibirá piezas a través de la cinta transportadora, la cual mediante carros distribuirá piezas a todos los robots de la estación.

Actualmente contamos con 4 plataformas que nos permiten ubicar las bandejas distribuidas por la cinta, en la siguiente imagen se indica la terminología para distinguirlas.

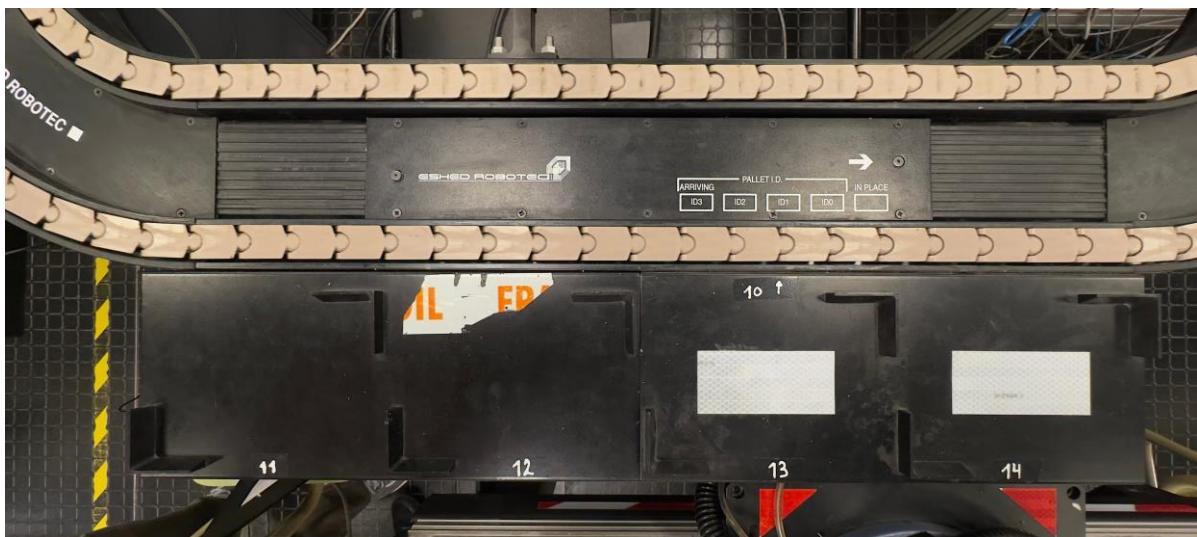


Imagen 2. Plataformas de trabajo

Circuito eléctrico del controlador

Se realizó el cableado desde el tablero "Manager" hasta el controlador del brazo para la comunicación entre ambas partes.

Como se muestra en la *imagen 3*, en el circuito se encuentra el cableado del gripper, y de las entradas y salidas con sus respectivos colores.

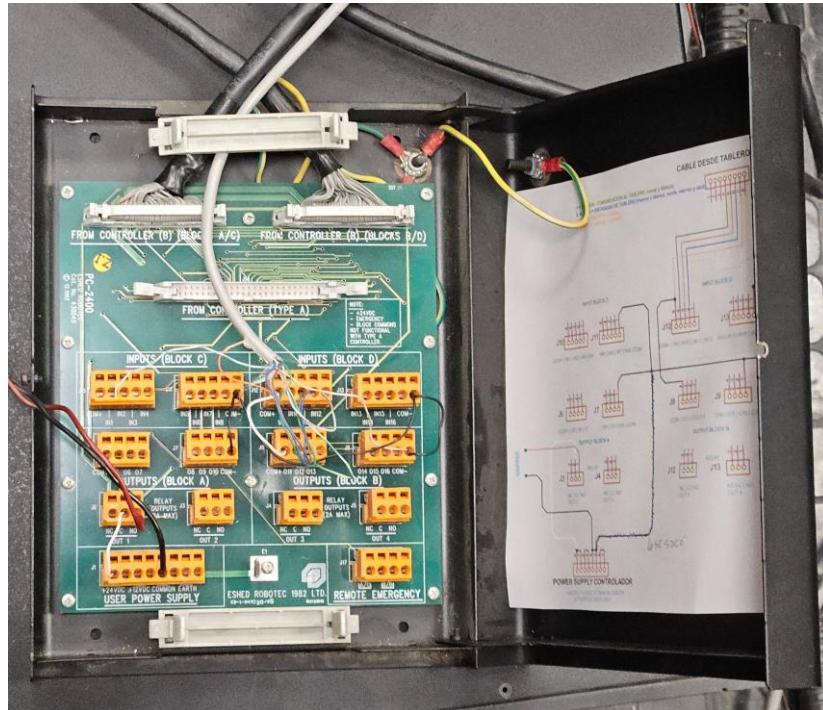


Imagen 3.1. Tablero del controlador junto a el diagrama

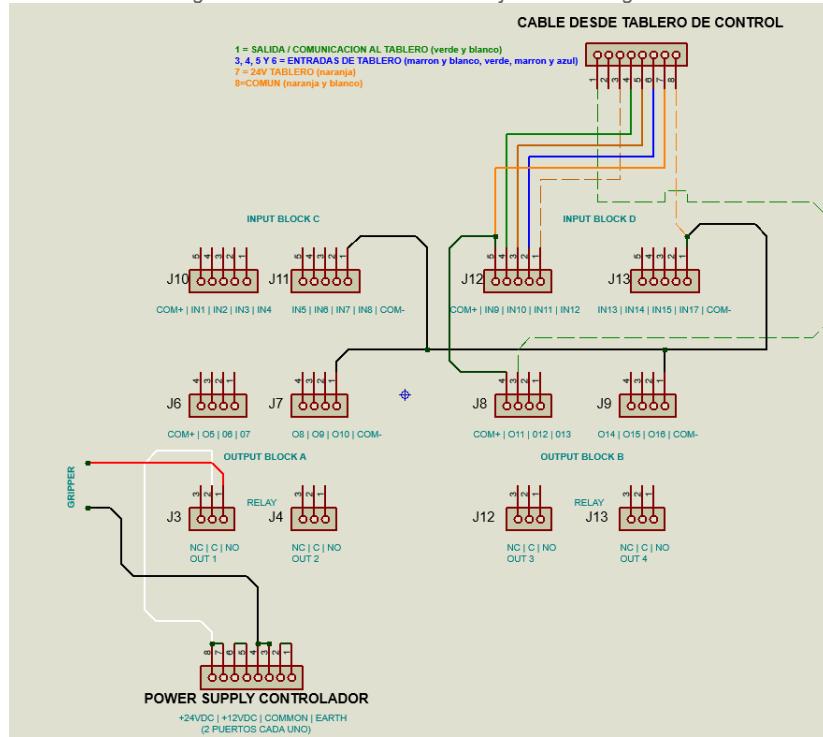


Imagen 3.2. Circuito eléctrico del tablero del controlador



PROGRAMACIÓN

Lenguaje

Para la programación del robot se utiliza principalmente el lenguaje ACL (*Advanced Control Language*). Utilizando el PC y Controlador como intermediario, se envían comandos a través del software ATS (*Advanced Terminal Software*). El mismo nos brinda un control total sobre el brazo robótico, permitiéndonos guardar posiciones, crear programas e interactuar con el Manager.

Este lenguaje funciona de forma similar al lenguaje de programación C, utilizando principios y funciones similares facilitando su uso.

Dentro del software ACL existen 2 modos de uso: DIRECT y EDIT, siendo el modo DIRECT el modo predeterminado permitiendo ejecutar comandos y realizar acciones con normalidad. Mientras que al modo EDIT se ingresa cuando se desea editar o crear un programa. Ambos permiten funcionalidades similares, existiendo comandos exclusivos a cada modo.

Al cambiar entre el control mediante el *Teach Pendant* y el modo Auto, es necesario escribir el comando “AUTO” para proceder a la programación.

Vectores

Para grabar posiciones del brazo robótico, es necesario crear vectores donde almacenarlas. Los vectores pueden definirse indicando su nombre y número de posiciones, en caso de no indicar su número de posiciones se definirá un vector de una sola posición.

Para crear un vector se utiliza el comando “DIMPA/B vect[n]”, donde:

- **A/B** define si el vector se creará para el brazo robótico (Grupo A) o para la base lineal (Grupo B).
- **vect** indica el nombre del vector a crear
- **[n]** es el número de posiciones que contendrá el vector.

Una vez creado el vector, se debe ubicar al brazo robótico en la posición que se desea guardar utilizando al *Teach Pendant*. Para guardar la posición se debe ingresar el comando “HERE vect[n]”, indicando con el valor “n” el índice del vector en la cual se desea almacenar la posición.

Luego se puede llamar a la posición del vector dentro de un programa para mover al brazo robótico a dicha posición.

Es necesario saber que la forma en la que se desplace el robot a dicha posición influirá en el vector, si se desplaza a la posición utilizando el modo joint, la posición se almacenará con movimiento articular. En cambio, si se desplaza hacia la posición con movimiento XYZ, la posición se almacenará con movimiento lineal. Es necesario saber esto para su posterior llamado dentro de un programa.

Programas

Es posible crear programas que ejecuten un ciclo determinado, a ser ejecutados manualmente o como consecuencia de una entrada real.

Para crear un programa se utiliza el comando “EDIT prog”, siendo “prog” el nombre que se le asignará al programa.

Al crearlo veremos una interfaz con líneas de comando numeradas, cada una de ellas nos permitirá introducir un comando a realizar, en secuencia.



Para ejecutar un programa se utiliza el comando “RUN prog”. Para abortar todos los programas en ejecución se utiliza el comando “A”.

Comandos y Recomendaciones

Se recomienda iniciar los programas definiendo las velocidades de cada articulación, siendo las articulares un porcentaje (0-100%) y las lineales en *mm/seg*. Para las primeras se utiliza el comando “SPEEDA/B var”, siendo

- **A/B** define si la velocidad afectará al brazo robótico (Grupo A) o a la base lineal (Grupo B).
- **var** es el valor porcentual de la velocidad, también puede colocarse una variable como veremos más adelante.

Es necesario también definir la velocidad lineal utilizando “SPEEDL var”

Luego continuamos con el programa como tal, para indicar al brazo que se desplace a una posición mediante movimiento articular, utilizaremos “MOVED vect[n]”. Es importante agregar la terminación “D” al comando para asegurarnos de que el programa no continuará hasta que el movimiento haya finalizado.

Si se desean realizar movimientos lineales, se utiliza el comando “MOVELD vect[n]”.

También contamos con comandos para controlar el estado del gripper todo/nada como “OPEN” y “CLOSE”.

Existe la posibilidad de crear ciclos convencionales dentro del programa tales como FOR e IF, también existen comandos conocidos como PRINT, PRINTLN y DELAY.

El estado de las salidas y entradas del controlador se puede verificar de dos maneras. Una es viendo la tapa frontal del mismo, que cuenta con luces que indican el estado de las mismas. Y la otra es utilizando el comando “SHOW DIN” para verificar entradas, y “SHOW DOUT” para salidas.

Para la lectura de entradas y salidas dentro de un programa se puede utilizar el comando “WAIT IN/OUT[n] =1/0”. Este comando detendrá el funcionamiento del programa hasta que la entrada o salida “n” se encuentre en el estado 1 o 0.

Para cambiar el estado de una salida, se utiliza el comando “SET OUT[n]=1/0” siendo “n” la salida a modificar y 1 / 0 el estado deseado.

Si se desea ejecutar un programa dentro de otro, es recomendable utilizar el comando “GOSUB prog”, el cual nos asegura que se ejecutará el programa por completo antes de avanzar a la siguiente línea. Pero también se puede ejecutar el comando “RUN prog” que permite ejecutar varios programas en simultáneo.

Variables

El lenguaje nos permite crear variables tanto globales como privadas, es importante saber que las variables solo pueden ser de tipo *int*, se definen de la siguiente manera:

- “DEFINE var” permite crear variables privadas
- “GLOBAL var” permite crear variables globales
- “DIM var[n]” permite crear un array de n variables privadas
- “DIMG var[n]” permite crear un array de n variables globales.

Para modificar el contenido de una variable se utiliza el comando “SET var=x” siendo x el valor que se desea asignar.



Programas y uso de la estación de trabajo

PROGRAMAS CREADOS

Para facilitar el uso del sistema a futuros estudiantes hemos creado una serie de programas fáciles de entender y modificar, a ser utilizados en conjunto con el Manager. Se definieron 2 variables (SPDA y SPDB) las cuales se encargan de controlar la velocidad articular del brazo robótico y la base linear respectivamente, estas permiten una modificación rápida en caso de ser necesario. Cabe aclarar que la velocidad lineal no permite variables, por lo que esta debe ser modificada manualmente dentro de cada programa.

A continuación, listamos los programas creados y su funcionalidad, para ver el código completo, dirigirse al anexo.

LOBBY

Se encarga de ejecutar los programas “PR1”, “PR2”, “PR3” y “PR4” en simultáneo. Una vez en ejecución, cada PR se encuentra en estado de espera de un pulso de entrada en los puertos 9, 10, 11 y 12 respectivamente. Una vez detectado el pulso correspondiente, se frena la ejecución del resto de programas. Por ejemplo: se ingresa por el puerto 9 un pulso (correspondiente al “PR1”), el programa procede con las líneas de código “STOP PR2”, “STOP PR3”, “STOP PR4” y “STOP LOBBY”. Luego, continúa con la orden de trabajo que le corresponda. Cada programa finaliza enviando una señal ACK de duración de un segundo por el puerto de salida número 11 para comunicar al tablero de control la finalización de la secuencia.

GETXX

Los programas GETXX se encargan de la recolección de las bandejas desde las estaciones indicadas. Por ejemplo: El programa GET10 se encarga de recoger de la estación 10 la bandeja, dejándola en la posición auxiliar (CIM2[1])

PUTXX

Los programas PUTXX, (*a ser ejecutados luego de los GETXX*) se encargan del depósito de las bandejas en las estaciones indicadas. Por ejemplo: El programa PUT10 se encarga de depositar una bandeja en la estación 10.

PR1, PR2, PR3 Y PR4

Cada uno de estos programas ejecuta una orden de trabajo en las estaciones 11, 12, 13 y 14. Una vez recibido el pulso correspondiente, se ejecutará una secuencia compuesta por los subprogramas GETXX y PUTXX, junto a un programa específico de cada estación que se encarga de recorrer los puntos de un vector para realizar la tarea designada. Estos programas pueden ser modificados para ejecutar una secuencia específica cada vez que se reciba una entrada, actualmente colocamos programas de ejemplo a modo de muestra

PR1 recibe la bandeja y ejecuta el subprograma “ENSAM” que recorriendo el vector “ENSAM[8]” ejecutará una rutina de ensamblado, devolviendo la bandeja antes de finalizar.



PR2 recibe la bandeja y ejecuta el subprograma “PRCAJ” que recorriendo el vector “CAJA [4]” ejecutará una rutina de ensamblado, devolviendo la bandeja antes de finalizar.

PR3 recibe la bandeja y ejecuta el subprograma “TORN2” que recorriendo el vector “TORN2 [8]” ejecutará una rutina de ensamblado, devolviendo la bandeja antes de finalizar.

PR4 recibe la bandeja y ejecuta el subprograma “TORN” que recorriendo los vectores “HTORN [3]” y “7TORN[5]” ejecutará una rutina de ensamblado, devolviendo la bandeja antes de finalizar.

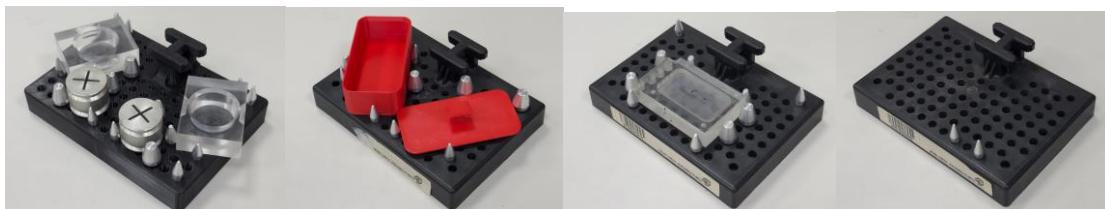


Imagen 4.1, 4.2, 4.3, 4.4. Bandejas correspondientes a los programas PR1, PR2, PR3, y PR4 respectivamente. En el estado precio a la ejecución de los mismos.

CONSIDERACIONES DE USO

Regulación de velocidad de cierre del gripper

En una de las patas del *Lineal Sidebase* se encuentra una válvula que se encarga de restringir el paso de aire al salir de la electroválvula del gripper. Esta nos permite regular la velocidad de apertura y cierre del mismo

Teach

Es necesario tener en cuenta que a la hora de manipular el robot con el uso del teach, el manejo de los ejes está asociados a diferentes grupos de control. Con respecto a los ejes del 1 al 5 (ya sean movimientos articulares o lineales) pertenecen al grupo A.

El eje extra, encargado del movimiento de la base lineal, pertenece al grupo B (eje 7). Se recomienda no presionar el botón del Eje 6 debido a que causará el congelamiento del controlador. Para descongelarlo se deberá presionar el pulsador de parada de emergencia.

Activación de ejes

Una vez reactivado el sistema luego de una parada de emergencia, los ejes deberán ser activados manualmente utilizando el TEACH de la siguiente manera:

- Verificar que el selector se encuentre en posición “TEACH”
- Presionar “CONTROL ON/OFF” hasta que figure en el display la leyenda “Con ALL”
- Presionar “ENTER” PARA CONFIRMAR

Encendido

Al encender el sistema, con el ejecutable del pc llamado “ATS”, se debe realizar el homing de las articulaciones del brazo, para esto se ejecuta el comando “RUN HOMES” el cual llama al programa “HOMES”.



Si los motores se encuentran bloqueados al encender el sistema, y al ejecutar el homing se lee el mensaje “NO SE HA EJECUTADO EL PROGRAMA SHUTD AL APAGAR, ANTES DE CONTINUAR ASEGÚRESE DE COLOCAR AL ROBOT EN UNA POSICIÓN SEGURA”, se debe ejecutar el programa “SHUTD” (Utilizando el comando “RUN SHUTD”).

En caso de que los motores sigan bloqueados, se recomienda el apagado y encendido de los motores manualmente desde el controlador presionando el pulsador con la leyenda “MOTORS”.

Apagado

Al finalizar el uso del brazo robótico, se debe realizar la ejecución del programa “SHUTD”, esperando hasta la finalización del mismo para el apagado. Si no se cumple, el próximo encendido se bloquean los motores.



Conclusiones

A nuestro parecer la realización de este proyecto fue desde principio a fin una experiencia enriquecedora, en la cual a través de desafíos únicos logramos poner a prueba nuestros conocimientos y superar los obstáculos encontrados.

Mediante la lectura del manual del brazo robótico, entendimos en profundidad el lenguaje utilizado, entradas, salidas, programadas, vectores, variables, ciclos, etc. Esto fue fundamental para el desarrollo del código principal “LOBBY” y todos los subprogramas que ejecuta.

Estamos agradecidos de haber formado parte de este gran proyecto. Esperamos haber logrado nuestro objetivo de facilitar el uso y seguimiento del sistema mediante la lectura de este informe.

Sin embargo, aún creemos que existe posibilidad de mejora del sistema, algunas de las tareas pendientes a realizar son las siguientes:

- Desarrollo de programas que realicen tareas colaborativas entre las diferentes estaciones de trabajo.
- Impresión 3D de nuevos componentes para interactuar.
- Implementación de botón de parada de emergencia.
- Calibración general de la estación para mejorar la alineación de los ejes cartesianos del brazo robótico.
- Realización de copias de seguridad de manera periódica del disco duro de la PC que contiene el software ATS.