Instalación ciros programming

C0.1. Archivos dentro de la carpeta “Ciros Programming 5.1.4”

Abrimos el setup como administrador, aparecerá esta pestaña:

Ciros programming

C1. Ventana iniciar al abrir “Ciros Programming”

File -> New -> Project Wizard

C2. Primera ventana del Project Wizard. Aquí se introduce el nombre del proyecto, el creador, una descripción y la ruta donde se guardará.

C3. Segunda ventana del Project Wizard. Aquí se introducen los parámetros del robot (tipo, cantidad de tarjetas I/O, cantidad de manos, ejes adicionales y lenguaje de programación).

Nuestro robot es RV2SDB, 2 I/O, 1 hand, MELFA-BASIC V, no additional axis

C4. Tercera ventana del Project Wizard. Este es un historial donde se escriben los cambios que se van realizando en el proyecto.

C5. Ventana principal del programa. Esta se divide en 5 ventanas, y cada una corresponde a un archivo diferente.

Previsualización -> Previsualización del robot. Está asociada a un archivo .mod

Pestaña mb5 -> En esta pestaña se escribe el código del robot. Está asociada a un archivo .mb5

Pestaña pos -> Se usa para guardar posiciones del robot. Está asociada a un archivo .pos

Pestaña explorer -> Principalmente para conectarse al robot.

Pestaña messages -> muestra mensajes de sistema (conectado al robot, errores, compilación...)

Es importante que todos los archivos se guarden dentro de la misma carpeta. También es importante que todos los archivos tengan el mismo nombre para el correcto funcionamiento del programa.

No es recomendable guardar los programas en la ubicación por defecto. Al usar el programa “ciros robotics”, hay una opción que involuntariamente puede restablecer esta ubicación y borrar el proyecto.

Podemos comprobar que nuestro código no de errores pulsando en “Programming -> Check syntax” o con el atajo ctrl+F9. Esto sacará por la ventana “Messages” un log indicando los errores que hay en el código. Estas indicaciones no siempre son precisas.

O0 -> piloto start

O1 -> piloto reset

O2 -> piloto lámpara Q1

O3 -> piloto lámpara Q2

O8 -> 1M1 – EVA de muelles

O9 -> 2M1 – EVA de tapas

I1 -> B1

I2 -> B2

I3 -> Pulsador Start

I4 -> Pulsador Stop

I5 -> Pulsador Reset

I6 -> Part\_AV – Pieza en rampa de entrada

I8 -> 1B1 cilindro 1M1 dentro

I9 -> 1B2 cilindro 1M1 fuera

I10 -> 2B1 cilindro 2M1 fuera

I11 -> 2B2 cilindro 2M1 dentro

I12 -> B3

I13 -> B1

I14 -> B2

En código, las entradas y salidas comparten número de bit (ambas van de 0-15). La variable se define como entrada o salida en función del primer uso que se le de en el programa.

C6. License manager

Poner la ip del servidor de licencias y conectarse por license service. Seleccionar la licencia y pulsar en generate configuration file. Tiene que estar esta referencia (fijarse en robotics.en):

Path del config file: C:\Program Files (x86)\didactic\CIROS Automation Suite 1.1\CIROS Robotics.en\bin\CIROS SmarxOS.ini

Ip pública del servidor:

Meteorologiajaumeprimer.com/meteo/reserva/temp/ip.txt

Ciros robotics

C7. Aparece al abrir el programa. Aquí se configura la comunicación con el robot.

Ir a la ruta C8. Aparecerá esta ventana (C9)

Abriremos el programa sin plc. Modelo de referencia es un programa base que no se puede variar.

Modelo de usuario hace una copia modificable del modelo de referencia. Borrar el modelo de usuario borra dicho modelo. Por eso no es recomendable guardar los programas en la ruta base.

Pulsamos en abrir modelo de usuario y, después de que termine de cargar, minimizamos la ventana para ir a la ventana principial del programa.

C10. Ventana principial del programa. Aparecerá una visualización 3D del robot, el código (archivo mb5) y las posiciones del robot (archivo pos) por defecto (los del model de referencia).

C11. En files eliminarlo todo y luego darle a “add files to project” para añadir los archivos mb5 y pos de nuestro usuario. Abrimos los dos archivos con doble click y al archivo mb5 le ponemos compile mode: main program.

Ahora ponemos en “compile Project” para subir el programa al robot. Si sale algún error, no habrá cargado el programa al robot y tendremos que arreglar dicho error para luego volver a compilar.

Damos ok al Project manager y volvemos a la ventana principal del programa. Aquí, para empezar la simulación, pulsamos en simulation -> start o con el atajo F5.

La función delay (dly) introduce un retardo en el programa. Durante la espera, el programa no hace caso a ningún input.