

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITÈCNICA SUPERIOR D'ALCOI

Automatización de una Bancada
Motorizada mediante PLC: Sistema de
Visualización

Trabajo Fin de Grado

Grado Ingeniería Eléctrica

Autor: Luke Thomas Nicolson

Tutor: Alfredo Villanova

Curso: 2014-15

Resumen

En este documento se va a explicar el funcionamiento de un taladro que se desplaza mediante una bancada automatizada que se puede mover en las coordenadas x e y, es decir, hacia la derecha o hacia la izquierda, todo a través de un control por autómatas Modicon 340 de la marca telemecanique, gobernado por un software de Schneider llamado Vijeo Designer V6.1 y Vijeo Webgate a través de cualquier pantalla táctil (HMI) de marca Schneider, también a través de la propia web o incluso con nuestro dispositivo móvil Android con la app de Schneider Design Air Plus.

Este proyecto contiene todo el diseño de los circuitos eléctricos y electrónicos, además de la programación del autómata programable a través de Unity Pro XL V8.0 enlazado con la pantalla diseñada vía Vijeo Designer y su conexión de Vijeo webgate a la red.

Tabla de contenidos

1. Objeto del proyecto	5
2. Motivación	5
3. Justificación.....	5
4. Proceso	5
5. Planteamientos.....	12
6. Exposición de las soluciones escogidas	15
6.1. Autómata programable y software	15
6.2. Protección de la instalación	17
6.2.1. Protección contra sobrecargas	17
6.2.2. Situación de los dispositivos de protección	18
6.2.3. Características de los dispositivos de protección.....	18
6.2.4. Cuadros de distribución.....	19
6.3. Justificación de los elementos utilizados	20
6.4. Principios de funcionamiento de los programas de control.....	31
7. Cálculos necesarios	35
8. Programas Unity PRO XL v8.0.....	37
8.1. Elementos del Unity PRO	37
8.2 Parámetros utilizados	42
8.3. Programa	43
8.4. Explicación del programa.....	54
8.4.1 Movimiento	54
8.4.2. Pentagrama.....	61
8.4.3. Esquema LD Puntos	62
8.4.4. Esquema SFC primario	63
8.4.5. Esquemas SFC secundarios.....	64
8.5. Pantalla Unity PRO XL v8.0.....	71
8.6. Creación de un programa con Unity PRO.....	72
9. Programa Vijeo Designer v6.1	79
9.1. Elementos de Vijeo Designer.....	79
9.2.Realización del proyecto	79
9.2.1. Gestión de los paneles.....	83
9.2.2. Paneles	84
9.3. Creación del programa	86
9.3.1. Añadir seguridad al control de la máquina	86

10. Vijeo Webgate	93
10.1. Creación del Webgate	93
11. Vijeo Design Air Plus	99
12. Esquemas.....	102
13.Referencias	105

1. Objeto del proyecto

Este documento tiene como objetivo la automatización de un taladro para poder hacer agujeros en cualquier tipo de superficie lisa

Aquí se mostrará el diseño de todo el proceso desde el control del autómeta hasta el mismo control vía internet o android de la máquina.

2. Motivación

La motivación de este proyecto es académica, ya que marca el final del Grado en ingeniería eléctrica, pero es también un reto personal de poder mostrar que puedo conseguir cualquier objetivo, y así demostrar lo aprendido durante estos últimos cuatro años, a la vez que presentarme en lo que podría ser una situación real.

3. Justificación

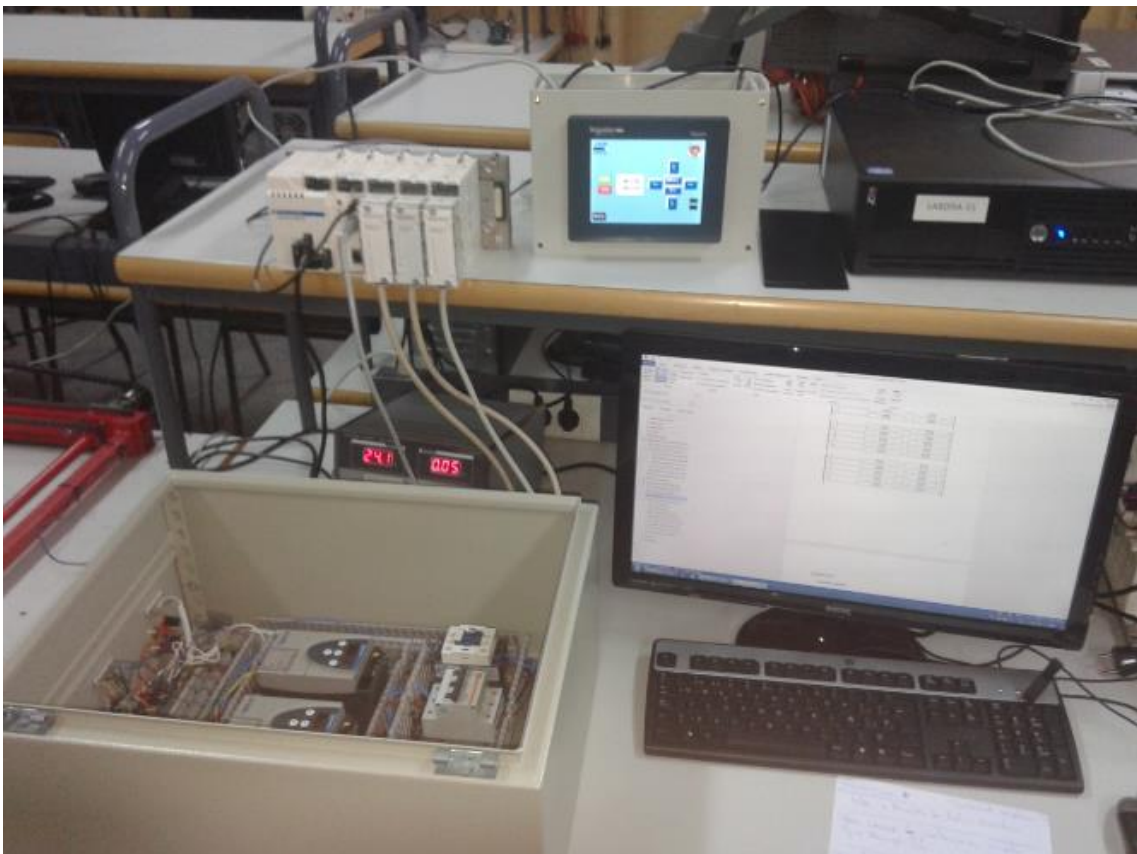
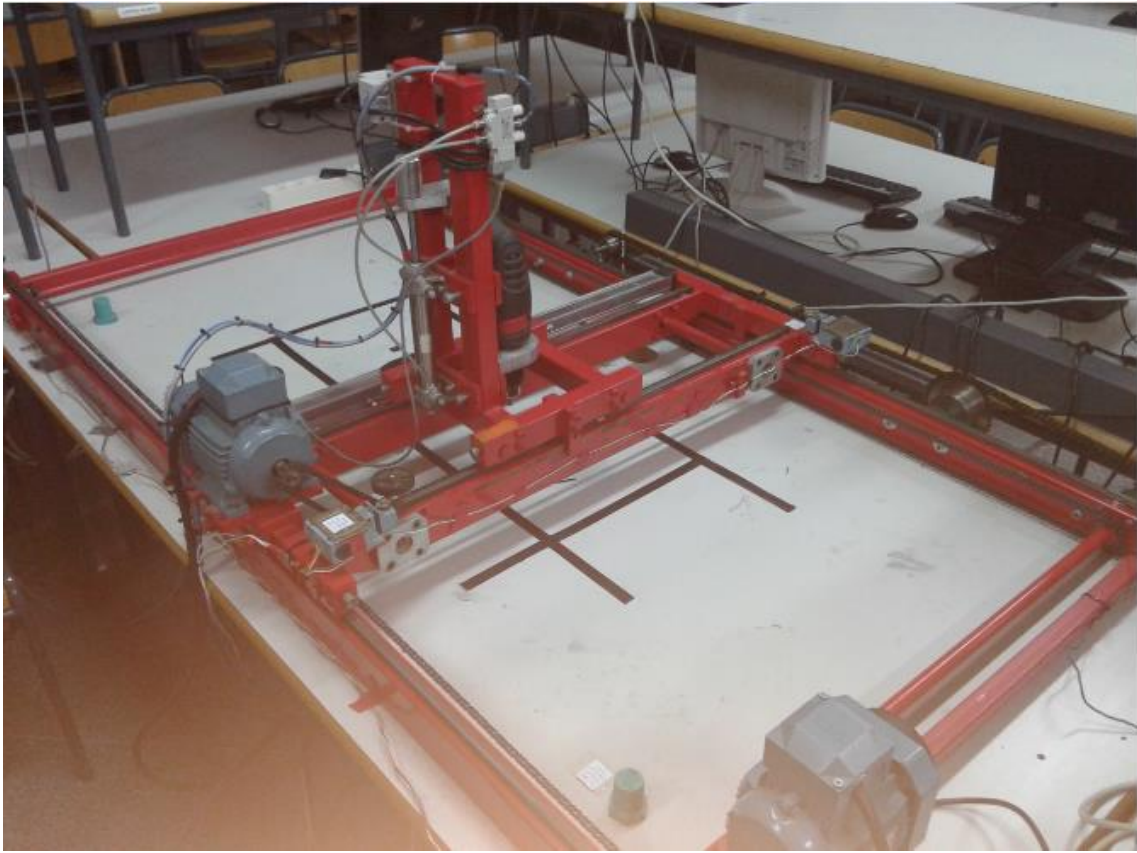
Este montaje se ha realizado por la disponibilidad debido a que está ubicado en el laboratorio F1L3 de la UPV del Campus de Alcoy, además de todas las facilidades que se han dado a la hora de necesitar autómetas, software, hardware y cualquier otro material que se haya necesitado durante todo el proceso.

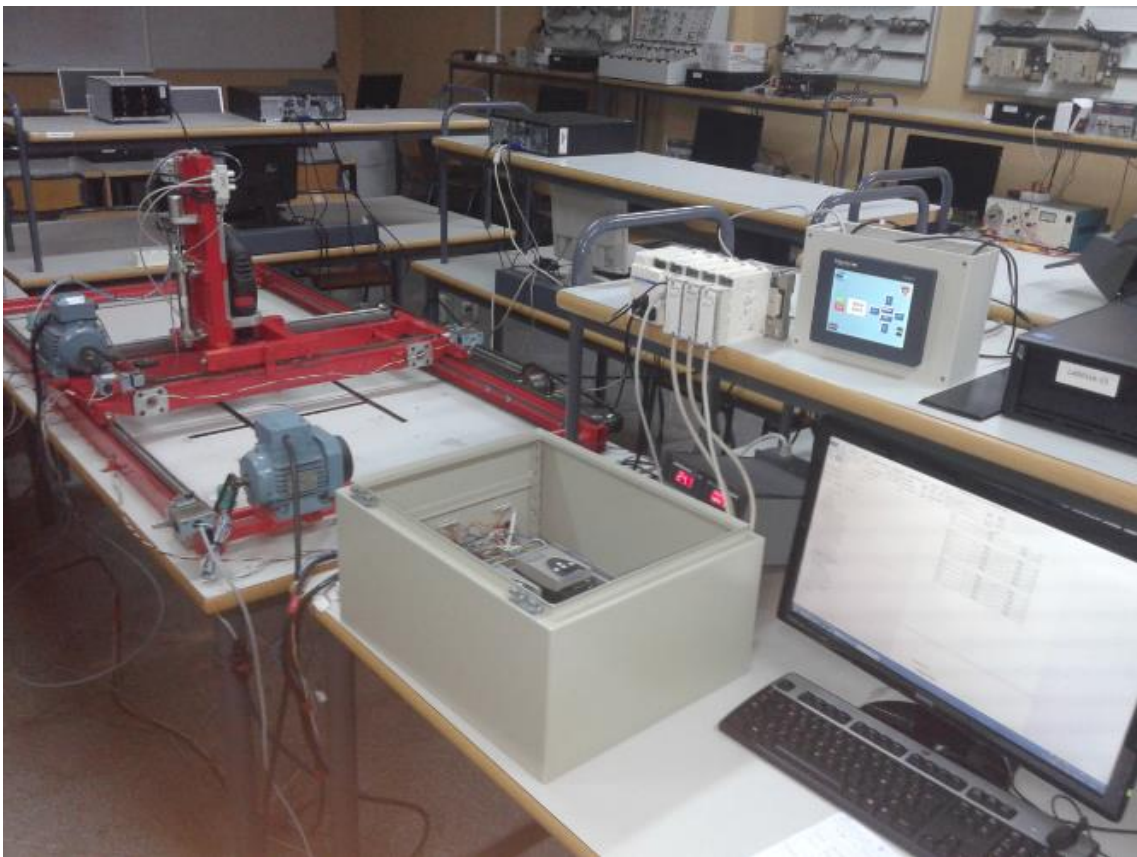
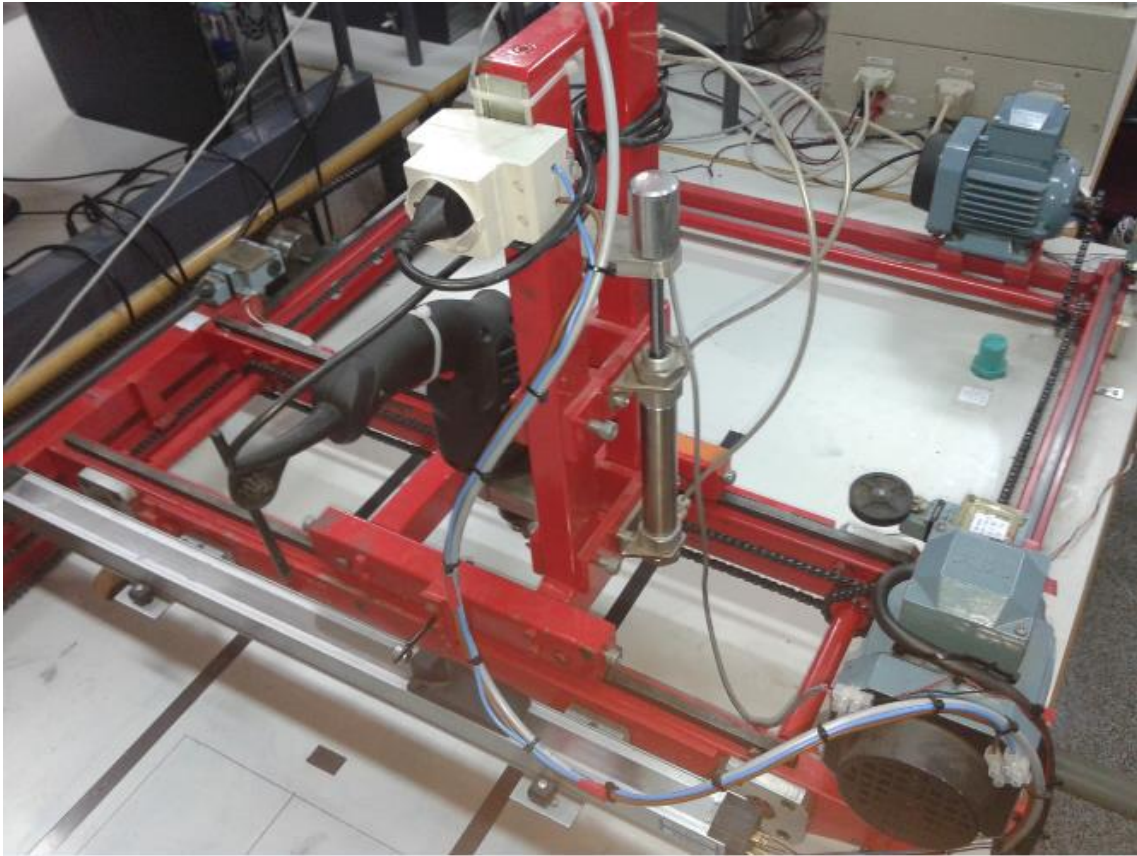
4. Proceso

La automatización no conlleva solo la automatización del proyecto, sino que se deben conocer todas las características y los datos técnicos de cada componente, aunque sean datos mecánicos, hay que conocer todos los detalles, por mínimos que sean todo influye.

Disponemos de dos carros de acero uno sobre el otro, un carro se mueve en el eje X y el otro carro se desplaza en el eje Y, mientras que hay un taladro que se desplaza en el eje Z a través de la acción de una electroválvula que mueve un pistón neumático.

Todo el sistema para mover la estructura está hecho con cadenas y engranajes conectados a motores para conseguir un desplazamiento entre sí. Se han empleado dos motores uno para el desplazamiento de derecha e izquierda, mientras que el otro manejará el desplazamiento hacia delante y hacia atrás





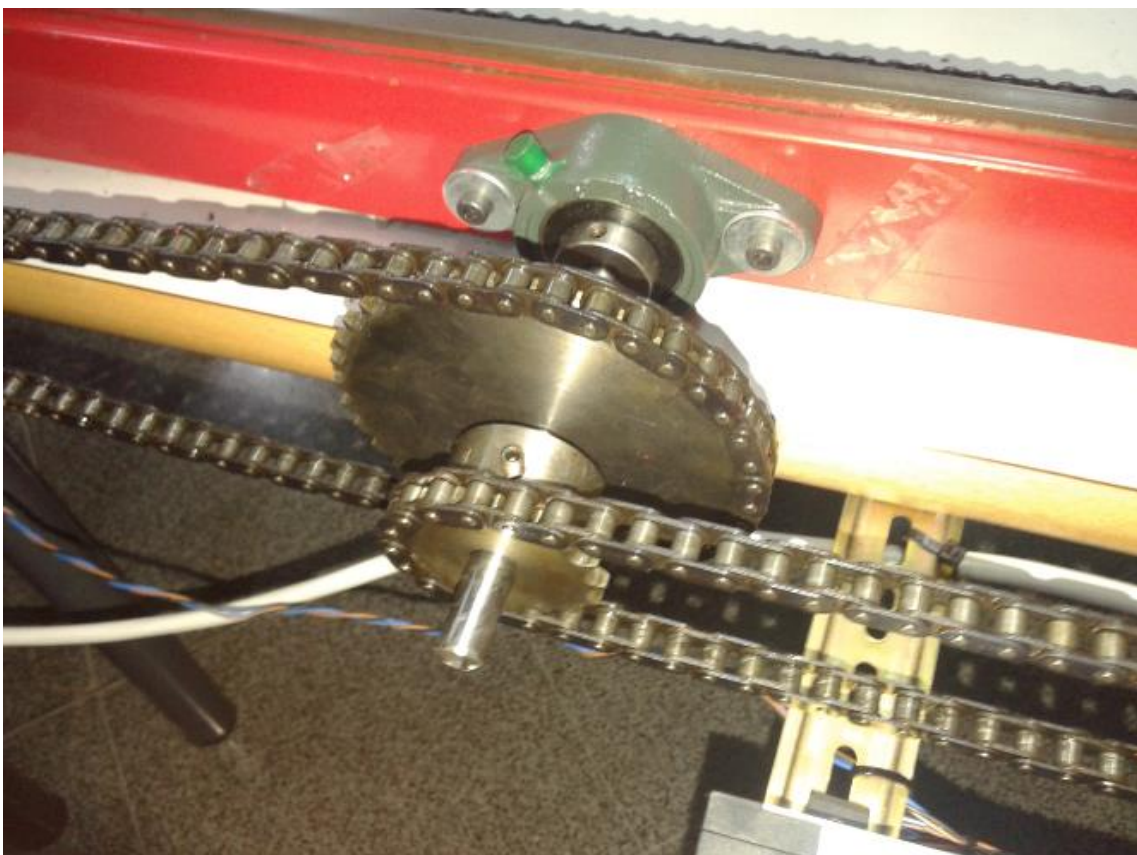
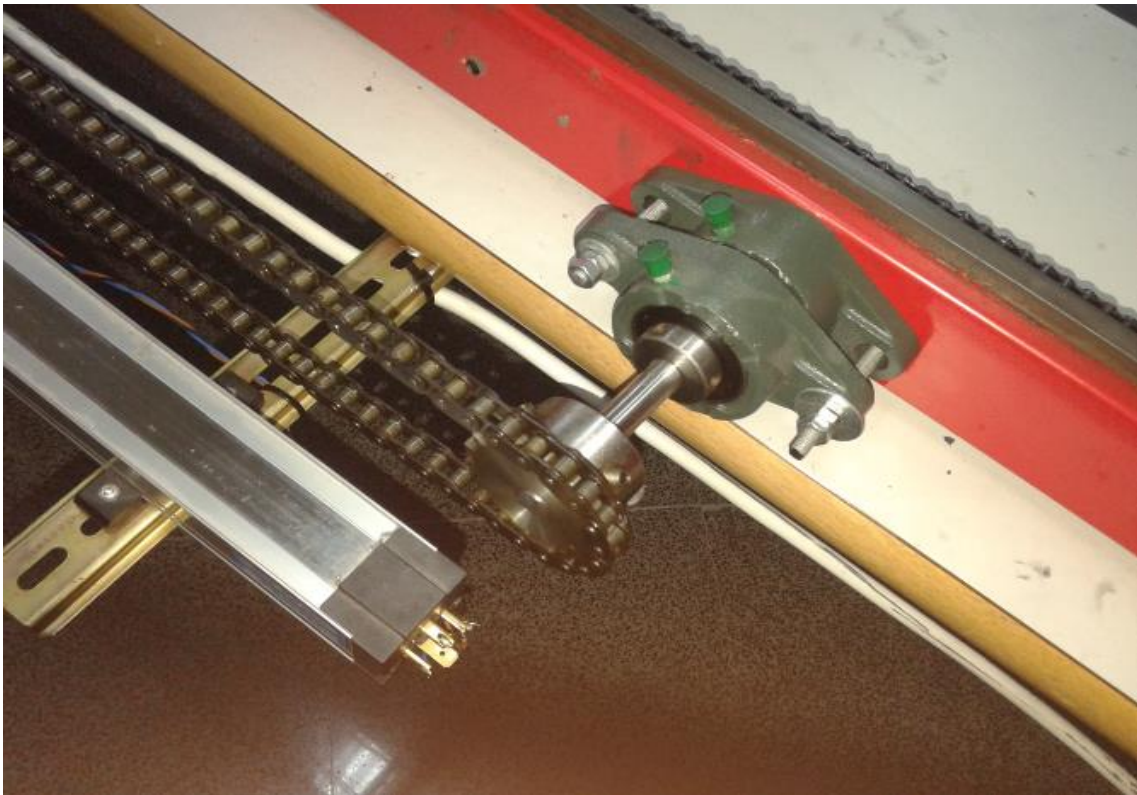
Las especificaciones técnicas que cumple la máquina son las siguientes:

Dimensiones:	1,8 x 0,9 m
Altura:	0,58 m
8 engranajes:	16 dientes
2 engranajes:	12 dientes
6 finales de carrera:	en los extremos de los ejes
2 motores trifásicos:	0,18 kW y $\cos \varphi = 0,64 - 0,68$
Máquina de taladrar:	500 W
Electroválvula:	0,15 – 0,7 MPa
Pistón Neumático:	1 Mpa

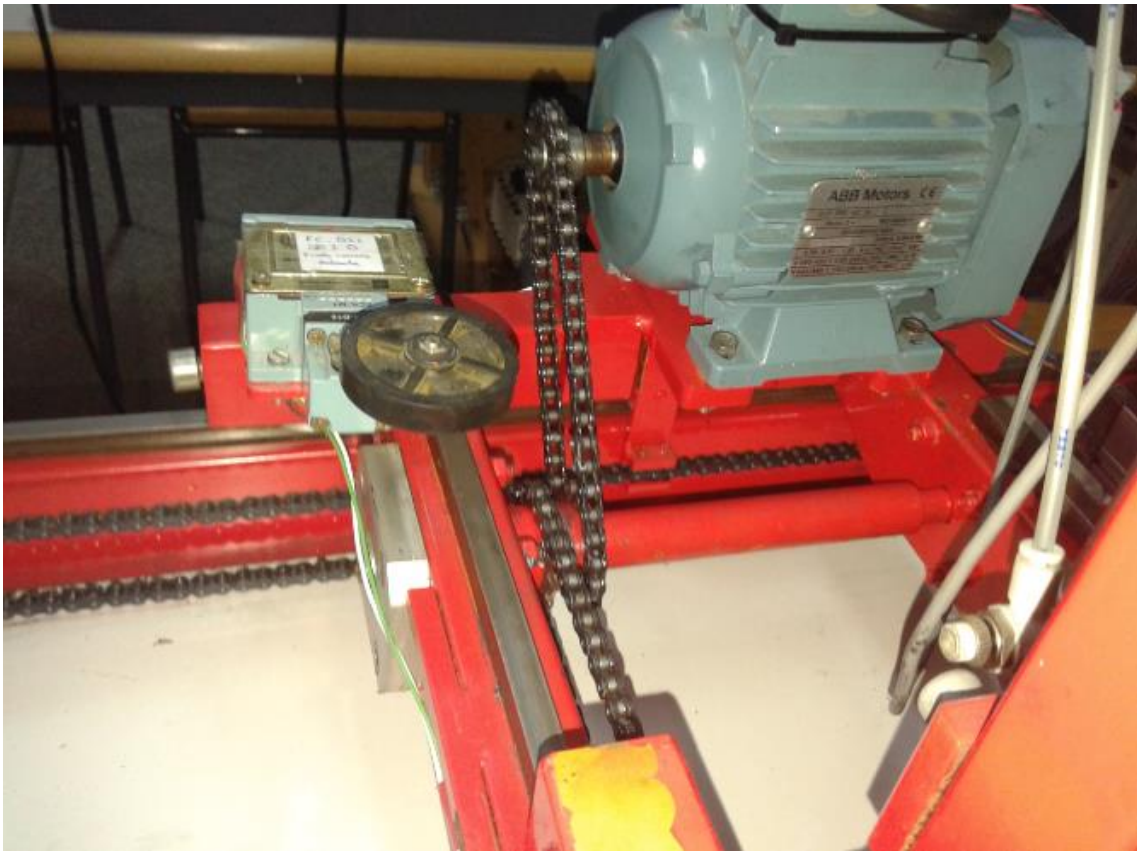


La gran mayoría de los elementos eléctricos se encuentran en un cuadro de un 0,4 x 0,5 m que contiene un contactor, un interruptor automático, dos variadores de frecuencia para los motores trifásicos y todas las regletas de conexión.

En las siguientes figuras se puede ver cada parte de la máquina.

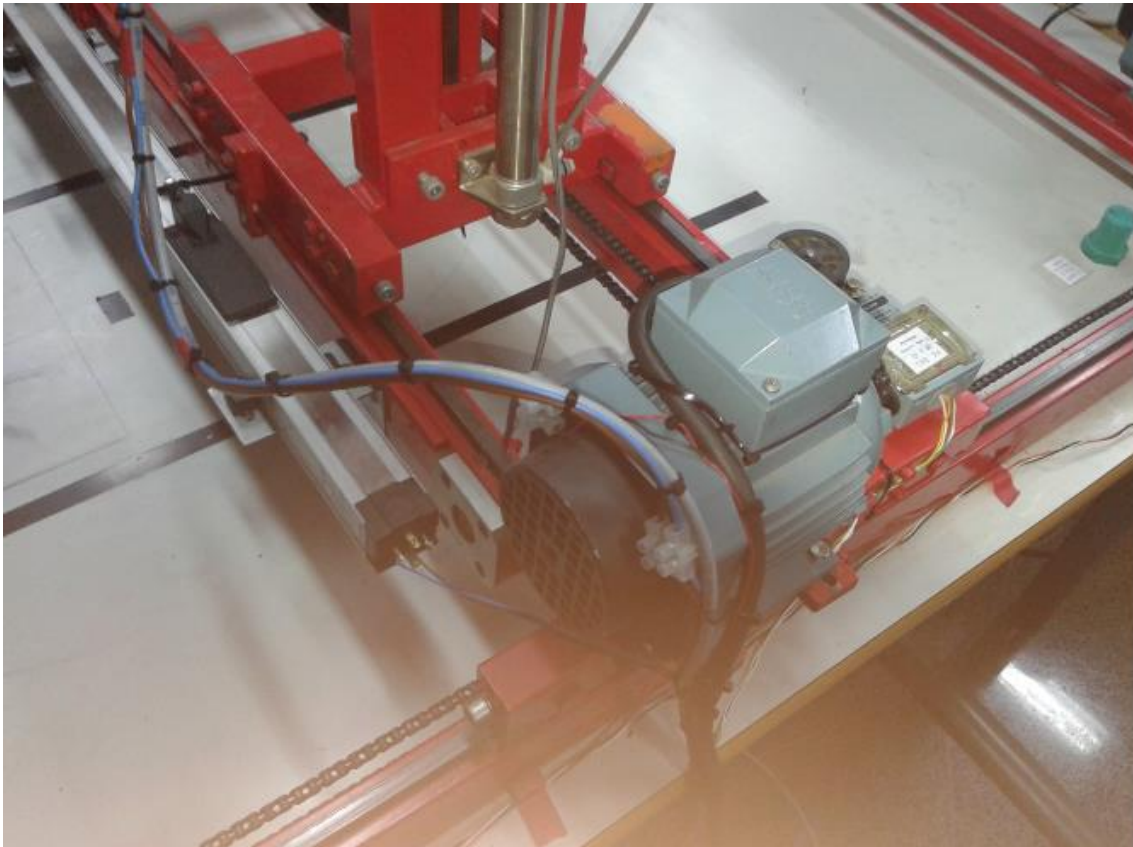






5. Planteamientos

El proceso consta de varias fases, puede taladrar en cualquier coordenada que se marque en pantalla, se puede manejar y desplazar manualmente y además realizar diferentes programas que ya tiene grabados como hacer cuadrados, pentágonos, rallas en diagonal, buscar el centro o incluso hacer un reconocimiento de su zona de trabajo para mantenimiento.



A continuación describiremos el principio de funcionamiento de cada una de las fases realizadas, y en los siguientes apartados se irán viendo las soluciones que hemos escogido.

El prototipo se ha creado para realizar la función de agujerear en cualquier coordenada que se le indique. El proceso se puede iniciar desde cualquier punto accesible al taladro.

Comenzará al activar la variable "IR", es una variable que se ha creado para ordenar a la máquina cuando comenzar el proceso automático, aunque también se ha creado otra variable "izq" para poder usar el modo de movimiento manual.

A partir de este momento, con la variable IR activada podremos navegar por el programa creado para situar la máquina en las coordenadas que insertamos o también utilizar los programas ya insertados para que realice diferentes

maniobras como perforar los agujeros de cada esquina de un cuadrado o pentágono, hacer agujeros en diagonal, buscar el centro de la tabla o incluso una opción de reconocimiento que se utilizaría en labores de mantenimiento de la máquina.

El funcionamiento consiste primero en apretar el pulsador Start para activar IR, y poder insertar los valores x e y deseados, automáticamente el taladro se colocará en la posición marcada, después de esto pulsando el botón Drill activaremos el giro del taladro y gracias a la acción de el botón EV se activará un pistón neumática que comenzará el descenso del taladro hasta la superficie a taladrar, la vuelta y mandar el taladro hacia arriba de nuevo se deberán activar apretando los botones nombrados.

Si activamos IR y el botón Manual ya podremos mover el taladro con las flechas de la pantalla, el uso del taladro es igual al anterior descrito.

Finalmente en la pantalla de Program podremos escoger diferentes programas ya insertados en el programa con unas coordenadas determinadas como Px y Py, con esto el taladro realizará:

-Cuadrado: El taladro tiene cuatro puntos marcados, nada más se active el botón con la figura de un cuadrado el taladro comenzará inmediatamente, irá al primer punto marcado, activará el taladro y acto seguido el descenso del taladro, una vez hecho esto, el taladro vuelve a subir se desactiva y pasa al siguiente punto marcado para volver a realizar el mismo movimiento en cada punto marcado hasta acabar en el último punto en espera.

-Pentágono: El taladro tiene cinco puntos marcados, nada más se active el botón con la figura de un pentágono el taladro comenzará inmediatamente, irá al primer punto marcado, activará el taladro y acto seguido el descenso del taladro, una vez hecho esto, el taladro vuelve a subir se desactiva y pasa al siguiente punto marcado para volver a realizar el mismo movimiento en cada punto marcado hasta acabar en el último punto en espera.

-Diagonal: El taladro tiene cuatro puntos marcados, nada más se active el botón con la línea en diagonal el taladro comenzará inmediatamente, irá al primer punto marcado, activará el taladro y acto seguido el descenso del taladro, una vez hecho esto, el taladro vuelve a subir se desactiva y pasa al siguiente punto marcado para volver a realizar el mismo movimiento en cada punto marcado hasta acabar en el último punto en espera.

-Centro: Este desplazamiento consiste en buscar el centro de la superficie a agujerear, aunque la acción del taladro será manual.

-Reset: Este botón se ha creado por futuras razones de mantenimiento, ya que gracias a este programa el taladro se mueve a cada esquina de

su zona de trabajo útil y finalmente al centro sin activar ni el taladro ni el pistón neumático.

Hay también un pulsador de emergencia, ya que la seguridad es muy importante, y además hay cuatro finales de carrera en cada esquina y dos más en la mesa que son los finales de carrera primarios en el programa, los cuatro nombrados antes solo son de extrema precaución por si ocurriera algún error grave que provoque un movimiento no controlado del prototipo, la función de todos es cortar la tensión en los variadores, desactivar el taladro, el pistón y a la vez desactivar el programa hasta nueva orden.

Nunca se deberá de manipular la máquina con tensión.

6. Exposición de las soluciones escogidas

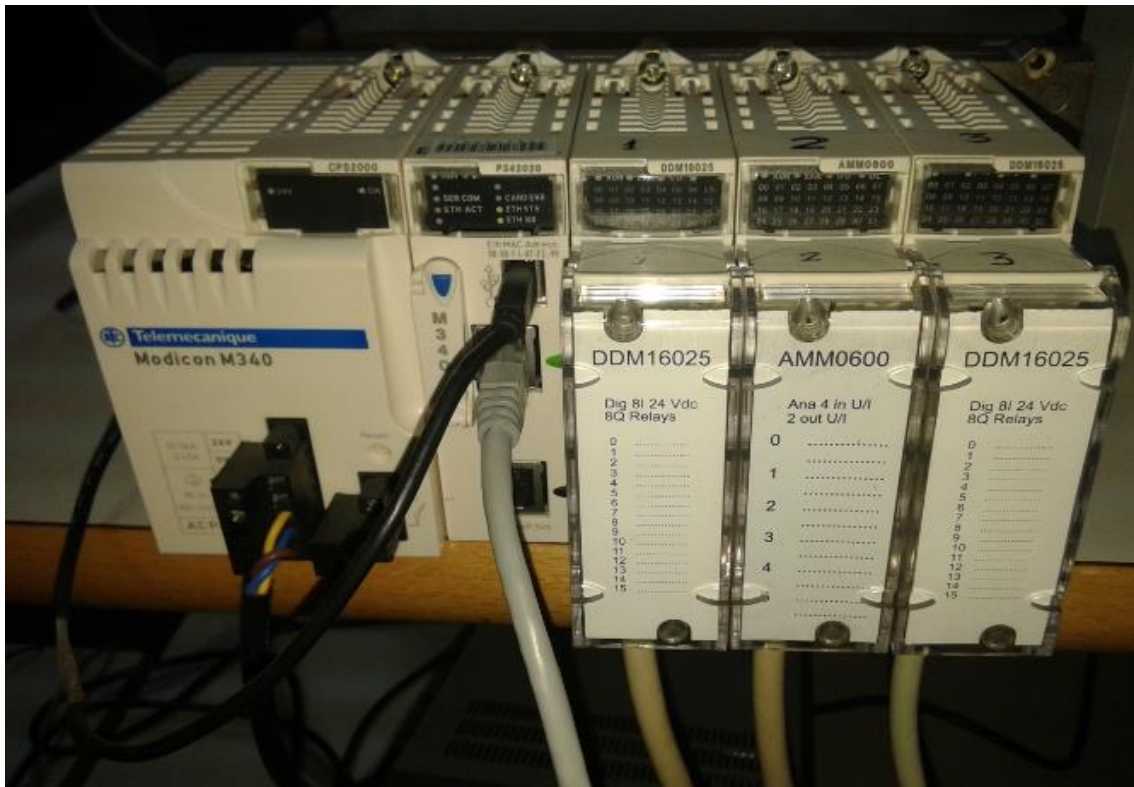
A continuación se analizarán las distintas opciones y soluciones adoptadas para la realización del proyecto.

Se diferencian las diferentes partes en la automatización:

1. Diseño de los circuitos de potencia
2. Diseño de los circuitos de mando
3. Diseño del programa del autómatas programable
4. Diseño del programa de control
5. Instalación de las máquinas

Los dos primeros apartados son parecidos. Los circuitos de potencia son los que normalmente se encargan de distribuir la energía entre receptores que necesitan potencias elevadas, ya que existen motores en la instalación. Los circuitos de maniobra manejan pequeñas tensiones, pero controlan todo el circuito de potencia.

6.1. Autómatas programable y software



- Facilidad:

Se ha utilizado un autómata programable Telemecanique modicon 340 con un módulo CP342020 con conexión a ethernet, un módulo AMM 0600, dos módulos DDM 16025. Todo el conjunto es compatible con el software de Unity de Telemecanique y Vijeo de Schneider, que nos permitirá total manejabilidad con un HMI, por la web o incluso con un dispositivo Android.

- Mayor longevidad de las instalaciones eléctricas

Con el uso de autómatas evitamos el desgaste físico de los contactos y así alargamos la vida de la máquina.

- Localización de errores

Gracias a las entradas digitales aumenta la posibilidad de encontrar fallos en todo el montaje y así será más sencilla la reparación

- Construcción robusta y de montaje sencillo

La caja base y todos los módulos de ampliación del autómata son bloques pequeños, manejables y robustos. No necesitan ventilación.

Los módulos de ampliación se enchufan en los elementos de bus, donde se atornillan a prueba de vibraciones. Los elementos de bus se enganchan sobre un carril normalizado.

- Inmune a interferencias

En una aplicación como la presente se necesita protecciones contra ruidos, interferencias y corrientes parásitas. En las fábricas donde hay aparatos de elevada potencia se emiten interferencias altas, entonces se necesita un dispositivo con inmunidad. La electrónica del dispositivo ofrece inmunidad frente a las interferencias.

- Actualidad

Se ha creado una pantalla de control con manejabilidad desde HMI, red y Android para adaptar la máquina al uso de software industrial actualmente usado en todo el mundo.

Con esto permitimos el uso de la máquina desde cualquier ubicación si fuese necesario.

6.2. Protección de la instalación

6.2.1. Protección contra sobreintensidades

Todo circuito estará protegido contra los efectos de las sobreintensidades que puedan presentarse en el mismo, para lo cual la interrupción de este circuito se realizará en un tiempo conveniente o estará dimensionado para las sobreintensidades previsibles.

Las sobreintensidades están motivadas por:

- Sobrecargas debidas a los aparatos de utilizados o defectos de aislamiento de gran impedancia.
- Cortocircuitos.
- Excepto los conductores de protección, todos los conductores que forman parte de un circuito, incluyendo el conductor neutro o compensador, estarán protegidos contra los efectos de las sobreintensidades.

1. Protección contra sobrecargas. El límite de intensidad de corriente admisible en un conductor ha de quedar en todo caso garantizada por el dispositivo de protección utilizado.

Para la protección del conductor neutro o compensador se tendrá en cuenta:

- Cuando el conductor neutro o compensador del circuito tenga una sección inferior a los conductores de fases o polares, y pueda preverse en él sobrecargas que no hagan actuar a los dispositivos de protección destinados exclusivamente a aquellos, se colocará un dispositivo de protección general que disponga de un elemento que controle en el conductor neutro o compensador, de forma que haga actuar el mismo cuando la sobrecarga en este conductor pueda considerarse excesiva.
El dispositivo de protección general puede estar constituido por un interruptor automático de corte omnipolar o por un interruptor automático que corte únicamente los conductores de fase bajo la acción del elemento que controle la corriente en el conductor neutro.
- En los demás casos, se admite que la protección del conductor neutro o compensador está convenientemente asegurada por los dispositivos que controlan la corriente en los conductores de fase. Como dispositivos de protección contra sobrecargas serán utilizados los fusibles calibrados de características de funcionamiento adecuadas o los interruptores automáticos con curva térmica de corte.

2. Protección contra cortocircuitos. En el origen de todo circuito se establecerá un dispositivo de protección contra cortocircuitos cuya

capacidad de corte estará acorde con la intensidad de cortocircuito que pueda presentarse en el punto de su instalación. Se admite, no obstante, que cuando se trate de circuitos derivados de uno principal, cada uno de estos circuitos derivados disponga de protección contra sobrecargas, mientras que un solo dispositivo general pueda asegurar la protección contra cortocircuitos para todos los circuitos derivados. Se admiten como dispositivos de protección contra cortocircuitos los fusibles de características de funcionamiento adecuadas y los interruptores automáticos con sistema de corte electromagnético.

6.2.2. Situación de los dispositivos de protección

En general, los dispositivos destinados a la protección de los circuitos se instalarán en el origen de éstos, así como en los puntos en los que la intensidad admisible disminuya por cambios debidos a sección, condiciones de instalación, sistema de ejecución o tipo de conductores utilizados.

6.2.3. Características de los dispositivos de protección.

Los dispositivos de protección cumplirán con las condiciones generales siguientes:

- Deberán poder soportar la influencia de los agentes exteriores a que estén sometidos, presentando el grado de protección que les corresponda de acuerdo con sus condiciones de instalación.
- Los fusibles irán colocados sobre material aislante incombustible, y estarán constituidos de forma que no puedan proyectar metal al fundirse. Cumplirán la condición de permitir su recambio bajo tensión de la instalación sin peligro alguno. Deberán llevar marcada la intensidad y tensión nominales de trabajo para las que han sido construidos.
- Los interruptores automáticos serán los apropiados a los circuitos a proteger respondiendo en su funcionamiento a las curvas de intensidad-tiempo adecuadas
Deberán cortar la corriente máxima del circuito en que estén colocados sin dar lugar a la formación de arco permanente, abriendo o cerrando los circuitos sin posibilidad de tomar una posición intermedia entre las correspondientes a las de apertura y cierre. Cuando se utilicen para la protección contra cortocircuitos, su capacidad de corte estará con la intensidad de cortocircuito que pueda presentarse en el punto de su instalación, salvo que vayan asociados con fusibles adecuados que cumplan este requisito. Los interruptores automáticos llevarán marcada su intensidad y tensiones nominales, el símbolo de la naturaleza de corriente en

que hayan de emplearse y el símbolo que indique las características de desconexión, de acuerdo con la norma que le corresponda, o en su defecto, irán acompañadas de las curvas de desconexión.



6.2.4. Cuadros de distribución

En el origen de toda instalación y lo más cerca posible del punto de alimentación a la misma, se colocará un cuadro de distribución en el que se dispondrán un interruptor general de corte omnipolar, así como los dispositivos de protección contra cortocircuitos y sobrecarga de cada uno de los circuitos que parten de dicho cuadro.

El cuadro de ha realizado con materiales homologados.



6.3. Justificación de los elementos utilizados

Contactores:

Una vez se ha conseguido realizar el circuito de maniobra de un motor, y se ha protegido el circuito de potencia del mismo. Se ha de disponer de un dispositivo de potencia que abra o cierre automáticamente la línea de alimentación del motor. Los contactores realizan dicha función. Se han utilizado en el proyecto los contactores de Telemecanique de tipo LC1D09 BD por las siguientes características:

1. Para controlar la activación del taladro.
2. Contactos auxiliares provistos de apertura forzosa para garantizar una vigilancia segura de los contactos principales.
3. Exento de materiales contaminantes y perjudiciales para la salud, como son el amianto y el cadmio.
4. Unos contactos auxiliares Normally Closed o Normally Open.
5. Posterior variación o ampliación de la composición de contactos auxiliares.

Interruptor automático trifásico:

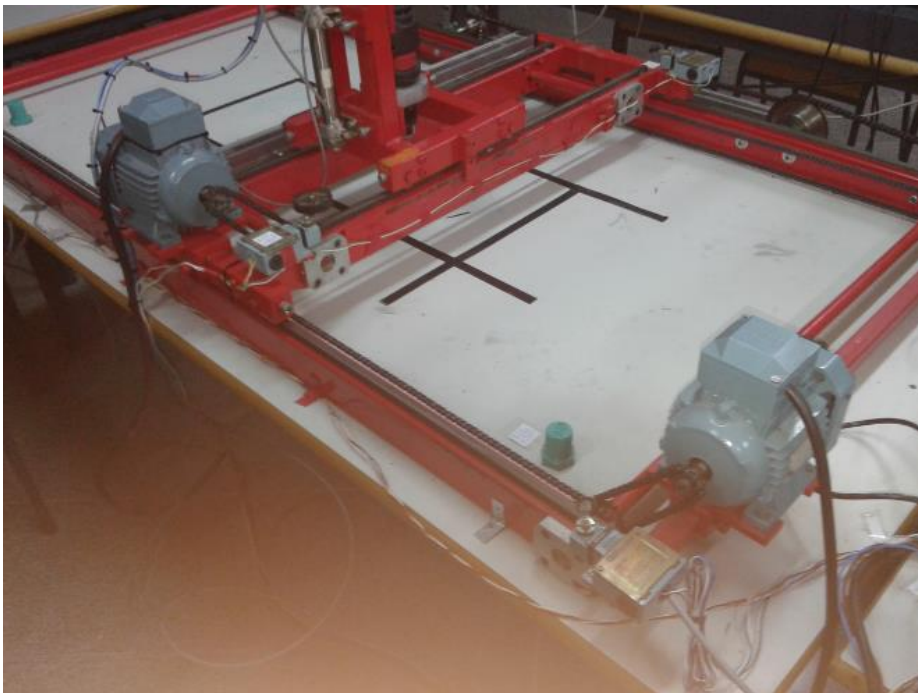


Un interruptor automático o disyuntor se emplea como dispositivo de protección es capaz de abrir un circuito magnético cuando la intensidad que por él circula excede de un determinado valor, para no causar daños a los equipos eléctricos, en este caso 63 A y una tensión nominal de 415 V.



Motores trifásicos:

Para el desplazamiento de la máquina de taladrar se ha utilizado 2 motores trifásicos, uno por cada eje. Se han conectado en estrella como se puede ver en la figura.



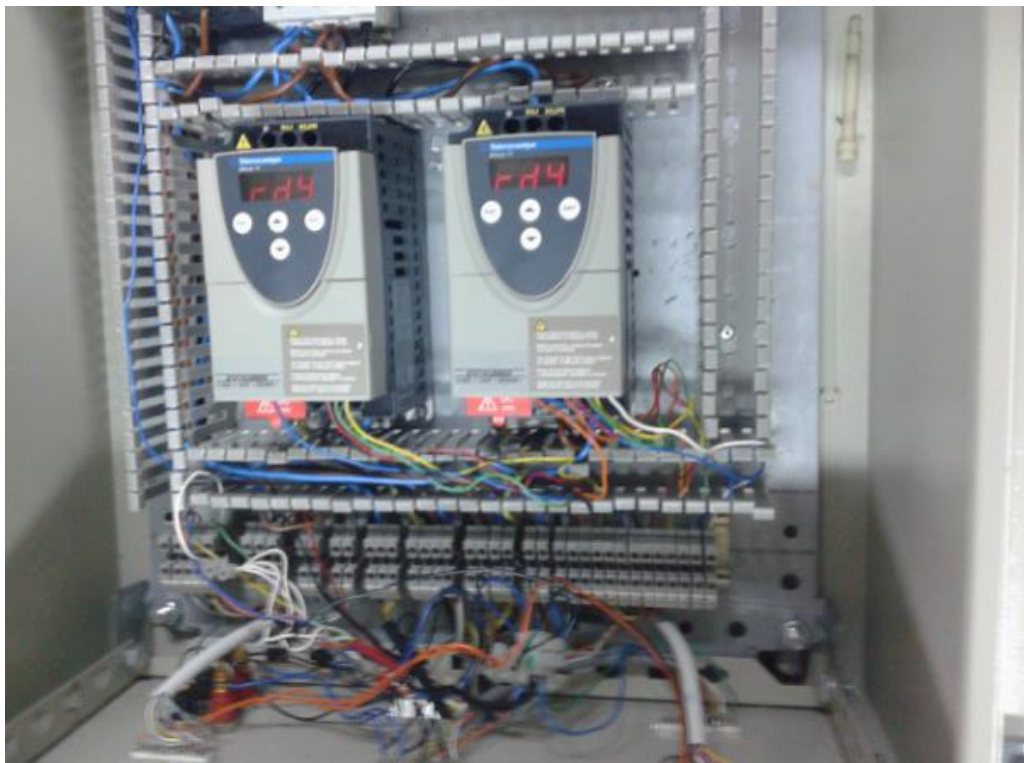
Se ha aprovechado los dos variadores de frecuencia para controlar el sentido del giro y también controlar la velocidad de los motores.

Máquina de taladrar:



Para agujerear se ha utilizado una máquina de taladrar normal y corriente de 500w, 50 Hz, 2,3 A y 3000 rev/min

Variador de frecuencia:



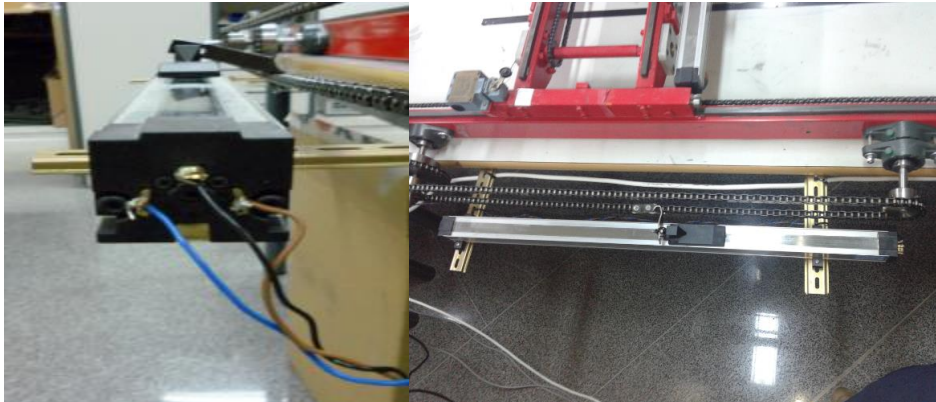
Pistón neumático:



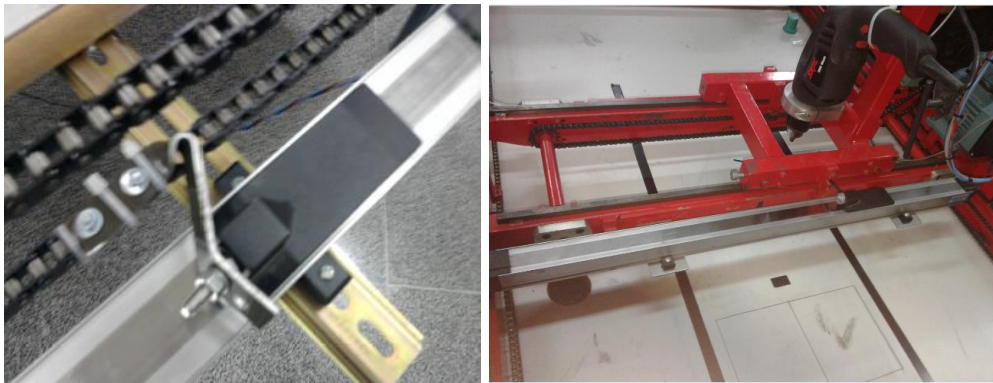
Se ha escogido un pistón de neumática SMC, para hacer subir y bajar la máquina de taladrar, mediante sus conductos por donde entra el aire a presión. Si entra el aire por el conducto de abajo el pistón se mantendrá arriba o si por el contrario entra el aire por el conducto de arriba el pistón bajará.

La presión máxima que puede soportar es de 1 M Pa, más que suficiente ya la presión que se proporciona es de 0,7 Mpa.

Sensor Lineal:



La función de estos es que el programa sepa la situación donde está situada la maqueta y estos limitan la zona donde el taladra no avanzará, estos siempre mostrarán en pantalla donde están en cada momento.



Electroválvula:

Una electroválvula es un dispositivo diseñado para controlar el flujo de un fluido a través de un conductor como puede ser una tubería. Tiene dos partes fundamentales: el solenoide y la válvula. El solenoide convierte la energía eléctrica mecánica para actuar la válvula.

Se ha utilizado este dispositivo de una entrada y dos salidas, para inyectar aire a presión por uno u otro conducto del pistón los cuales hacen subir o bajar el pistón de neumática. Se ha elegido una electroválvula SMC SY5120-5LOU-01-Q, la cual tiene una entrada por donde entra el aire a presión y dos salidas, la de arriba para subir el pistón (por defecto) y la otra para bajarlo, cuando se activa el solenoide. Soporta unas presiones entre 0,15 y 0,7 MPa. El solenoide se activa a 24V de continua, lo cual hace cambiar de conducto de salida.



Autómata programable Modicon 340:



Es una CPU todo en uno. Tiene sistema multitarea para tiempo rélfex garantizado. Tiene puerto USB para programación integrado en todas

las CPUs y 2 puertos adicionales como desee: Ethernet, CANopen, El modbus es extremadamente compacto, el nuevo autómata Modicon M340 incluye toda la flexibilidad y los servicios de un autómata de gama alta. En el centro de la aplicación, ofrece soluciones integradas de Plug&Work con otros dispositivos Telemecanique. Directamente sobre su escritorio, la gran capacidad de la oferta Unity facilita y reduce el tiempo de programación. Modicon M340 es realmente el pequeño gigante. Se aplica a máquinas complejas (empaquetado, textil, tratamiento de material, equipos elevadores...), máquinas de fabricación, de infraestructuras y es un microautómata.

Pantalla HMI:



Las nuevas terminales Magelis STO y STU se adaptan a sus necesidades ofreciéndole mayor flexibilidad, mayor capacidad de comunicación y un revolucionario sistema de montaje rápido y fácil.

- Vijeo Designer , el software de configuración fácil y potente
- Alimentación de 24 VDC
- Puertos seriales RJ45 ,RS485/RS232 multiprotocolos
- Puerto mini-USB, lista para usar memoria USB, teclado y más...

Magelis STU

- Pantalla a color de alta definición TFT – QVGA
- Disponible en dos tamaños de pantalla:

- HMISTU655(3.5") o HMISTU855 (5.7")
- Montaje tipo botón 22 mm Schneider Electric
- Puerto Ethernet RJ45 10/100 Base T
- Acceso remoto vía el explorador de Web Gate.

- Montaje revolucionario con los clips Magelis Spring para el STO y un sistema de botón para el STU
- Equipadas para comunicarse con equipos Schneider Electric y terceros
- Interfaz de operador de alto rendimiento y a precio accesible

Los módulos continúan alimentándose a 24 Voltios los digitales y a 5 Voltios los analógicos. Para esto se ha usado una fuente de alimentación de tensión continua.

Finales de carrera:

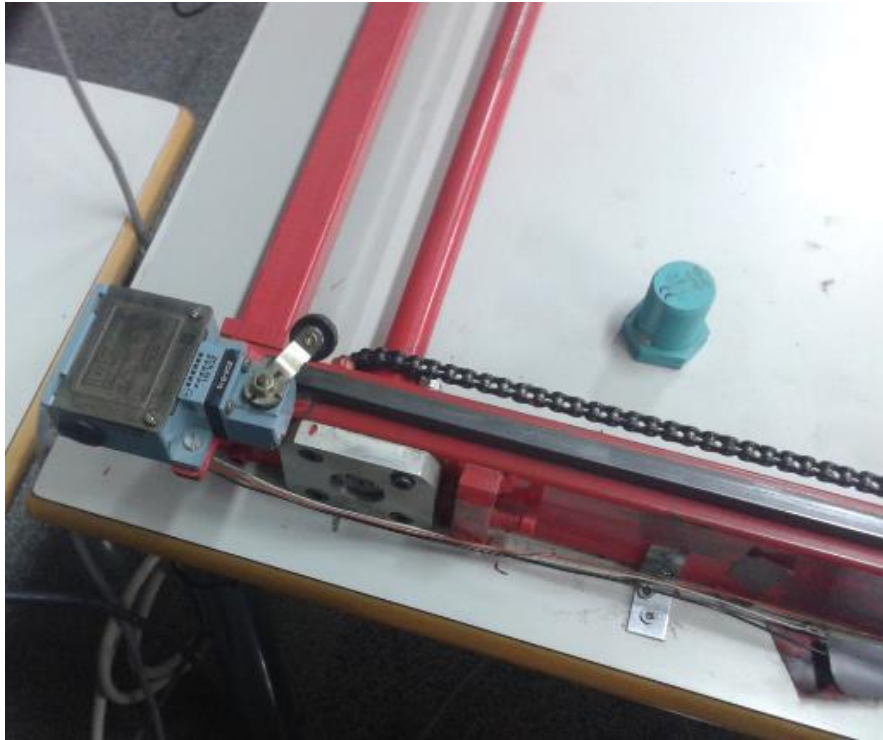
La función de los finales de carrera también conocidos como detectores electromecánicos, es detectar un objeto, en nuestro caso alguno de los carros, mediante el contacto físico.

Consisten en un cabezal que es accionado por el objeto, haciendo conmutar los contactos eléctricos del final de carrera, detectando así su presencia. Está formado por tres elementos:

1. Dispositivo de ataque: Actúa sobre el grupo de contactos y puede ser de diversos tipos como rodillos, varilla, rodillo extensible, varilla elástica, etc...en este caso se han utilizado rodillos.
2. Cabezal de mando: Aloja el soporte mecánico del dispositivo de ataque.
3. Cuerpo: Contiene los contactos del interruptor.

Su principal característica es su resistencia a impactos y vibraciones. Los finales de carrera elegidos han sido telemechanique XCK-M. Los cuales soportan una intensidad de 3 A y una tensión de 500 V. Es posible conectarlos como NC o como en el presente proyecto NO. Su función es detectar los extremos de cada eje.





Compresor:



Un compresor disminuye el volumen de una determinada cantidad de aire y aumenta su presión por procedimientos mecánicos. El aire comprimido posee una gran energía potencial, ya que si eliminamos la presión exterior, se expandiría rápidamente. El control de esta fuerza expansiva proporciona la fuerza motriz de muchas máquinas de herramientas.

Por eso se ha utilizado un compresor de la marca JOSVAL en este proyecto, para proporcionar la presión necesaria, en este caso 0,7 y 0,5 MPa, al circuito de neumática, el cual tiene como finalidad hacer subir y bajar el pistón.

Tubos neumáticos:

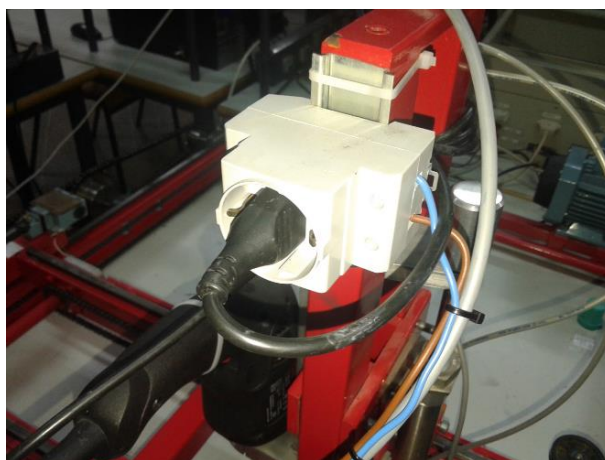
Se han usado tubos neumáticos FESTO de dos secciones diferentes. La sección más gruesa se usa para la salida del compresor de 6 x 1 y la más fina de 4 x 0,75 para la entrada de la electroválvula, se ha utilizado reductor neumático para reducir la sección del tubo.

Regletas:

Se han utilizado dos regletas, las cuales se ha atornillado al motor utilizando los tornillos que sujetan el ventilador, como se puede observar en la figura se ha utilizado para proporcionar 24 V de corriente continua a la electroválvula y la otra para los 230 V de corriente alterna que necesita la máquina de taladrar.

Soportan una tensión de 400 V, una corriente de 10 A y una temperatura de 85°C.

Enchufe:



Con la finalidad de conectar la máquina de taladrar se ha utilizado un enchufe de la marca Hager, para que sea más práctico y permita cambiarla por otra de diferentes características.

6.4. Principios de funcionamiento de los programas de control

El programa del PC considera los registros del autómata programable como suyos propios. De esta forma solamente tiene que leerlos y actuar en consecuencia. Si se necesita escribir en ellos, se procede de igual forma. En realidad son tratados como simples variables.

Tanto el programa del autómata como el PC se basan en la ejecución de instrucciones de forma secuencial, esto es, una vez finalizado el recorrido completo de programa, modifica las salidas y posteriormente comienza nuevamente su primera instrucción.

El autómata consulta los estados de señal de todos los módulos de entrada (digitales y analógicos), y forma la imagen de proceso de las entradas, denominada PAE.

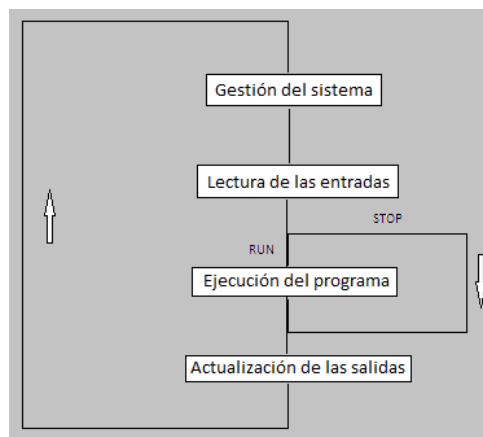
Una vez ha realizado esta copia de las señales de entrada, comienza el procesamiento del programa. La unidad de control toma de la memoria interna del programa (instrucción por instrucción), y las ejecuta. Para ello establece combinaciones, efectúa cálculos con los datos de la PAE y considera los estados de los temporizadores, contadores, otros bloques de función, así como las marcas.

La unidad de control del autómata deposita el resultado del procesamiento del programa en la imagen del proceso de las salidas PAA.

La unidad de control del autómata solo transfiere a los módulos de salida, temporizadores y contadores, los estados de señal contenidos en la imagen de proceso de las salidas PAA cuando ha finalizado el

recorrido del programa, esto es, al final de un ciclo. Ahora puede comenzar de nuevo el ciclo.

La programación mediante Unity XL PRO v8.0 del autómatas es totalmente estructurada y utiliza una estructura mono tarea. En la figura se aprecia todo lo explicado anteriormente.



La forma de representación escogida para programar el autómatas ha sido LD y SFC, es decir Ladder Diagram and Sequential Function Chart. Ya que es una forma sencilla y muy visual.

En Unity el editor LD posibilita la programación gráfica del ladder según IEC 1131-3.

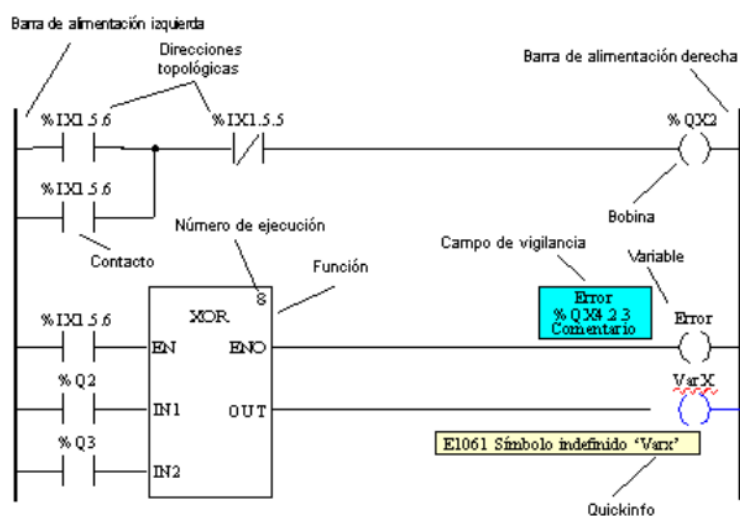
La estructura de un programa LD equivale a un circuito de corriente para conmutadores de relé.

En el lado izquierdo del editor LD se encuentra la denominada barra de alimentación izquierda. Esta barra de alimentación izquierda corresponde a la fase (conductor L) de un circuito de corriente. Al igual que en un circuito de corriente, en la programación LD sólo se editan los objetos LD que estén conectados a una fuente de alimentación, es decir, que estén conectados a la barra de alimentación izquierda. La barra de alimentación derecha equivale al conductor neutro.

Las propiedades de un programa LD son:

- Una sección LD tiene un ancho de entre 11 y 64 columnas y entre 17 y 2000 filas. (se pueden cambiar).
- Los programas LD están basados en celdas, es decir, en cada celda se puede colocar un único objeto.

Representación de un programa LD:

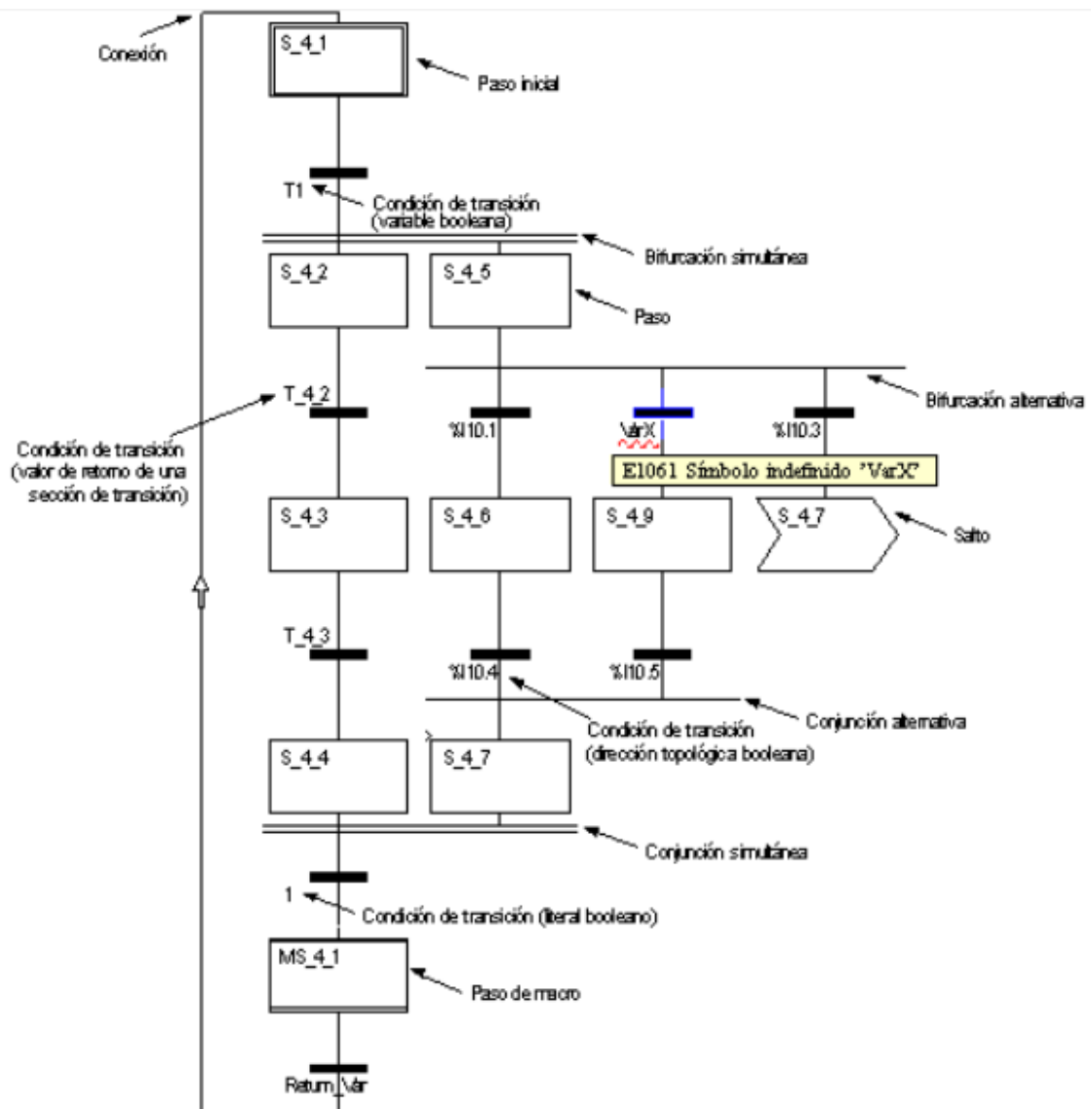


Con Unity también tenemos el editor SFC que es un control de secuencia compuesto por secciones de transición y secciones de acción. Las secciones SFC sólo se admiten en la tarea master del proyecto. Cada sección SFC contiene uno o más escalones SFC (cadenas secuenciales). Una sección SFC está compuesta por una ventana con una sola página. La ventana tiene una rejilla lógica de fondo con 200 líneas y 32 columnas.

Sus propiedades son las siguientes:

- Los pasos, las transiciones y los saltos necesitan una celda para cada uno.
- Las bifurcaciones y las conjunciones no necesitan disponer de su propia celda, sino que se insertan en la celda del paso o de la transición correspondiente.
- Por cada sección SFC (incluyendo todas sus secciones de marco) se puede ubicar un máximo de 1024 pasos.
- Por cada sección SFC (incluyendo todas sus secciones de macro) puede haber un máximo de 100 pasos activos (Multi-Token).
- Por cada sección SFC se puede activar al mismo tiempo un máximo de 64 pasos (Multi-Token).
- A cada paso SFC se le puede asignar un máximo de 20 acciones.

Representación de una parte de sección SFC:



7. Cálculos necesarios

En este apartado veremos los cálculos que se han realizado para saber que componentes de la maqueta escoger.

Intensidad de línea de los motores trifásicos

La expresión (1) define la intensidad en un circuito trifásico

$$I_l = \frac{P}{\sqrt{3} \cdot V_{ll,RMS} \cdot \cos\varphi} \quad (1)$$

$$I_l = \frac{0,18 \cdot 10^3}{\sqrt{3} \cdot V_{ll,RMS} \cdot \cos\varphi} \quad (2)$$

Siendo:

P = potencia que se transporta en vatios

$V_{ll,RMS}$ = Valor eficaz de la tensión entre fases en voltios

$\cos\varphi$ = Factor de potencia del motor

Potencia total consumida

Las potencias totales de las máquinas vienen dadas por las potencias parciales que consumen sus respectivos motores, en el presente proyecto se han utilizado 3 motores:

Dos motores trifásicos de la misma potencia y el de la máquina de taladrar que es monofásico.

$$P_t = P_{motor\ horizontal} + P_{motor\ vertical} + P_{taladro} \quad (3)$$

$$P_t = 180 + 180 + 500 \quad (4)$$

Relación de transmisión y velocidad del engranaje conducido

$$R_t = \frac{Z_1}{Z_t} \quad (5)$$

$$R_t = \frac{12}{16} = \frac{3}{4} = 0,75 \quad (6)$$

$$\omega_1 = 1380 \text{ rev/min} = 144,51 \text{ rad/s} \quad (7)$$

$$\omega_2 = \omega_1 \cdot R_t = 144,51 \text{ rad/s} \cdot 0,75 = 108,38 \text{ rad/s} \quad (8)$$

Siendo:

Z_1 = Número de dientes del piñón conductor.

Z_2 = Número de dientes del engranaje conducido.

R_t = Relación de transmisión.

ω_1 = Velocidad angular del motor y por tanto del piñón conductor en rad/s.

ω_2 = Velocidad angular del engranaje conducido en rad/s.

Velocidad del carro

Se ha utilizado la siguiente expresión (9) para calcular la velocidad lineal teórica que alcanza el carro. Transformando la velocidad angular del motor en lineal, es una aproximación, ya que no se ha tenido en cuenta muchos parámetros reales como el rozamiento, la inercia...

$$v = r \cdot \omega_2 \quad (9)$$
$$v = 0,017 \cdot 108,38 = 1,84 \text{ m/s} \quad (10)$$

Siendo:

R = El radio del engranaje en metros.

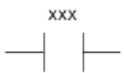

8. Programas Unity PRO XL v8.0


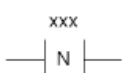
8.1. Elementos del Unity PRO

En este apartado se va a mostrar los elementos del editor LD y SFC utilizados para realizar el proyecto y también algunos otros muy importantes.


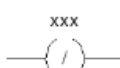

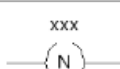
Los elementos se pueden ver en la siguiente figura, y los bloques de función existentes también.

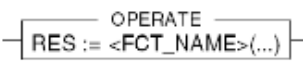
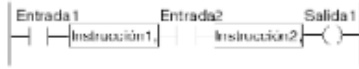
Elementos LD Tipos de contactos

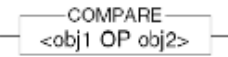
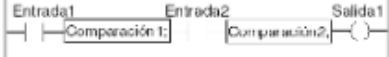
Designación	Representación	Descripción
Normalmente abierto		En el caso de los contactos normalmente abiertos, el estado de la conexión de la izquierda se transfiere a la conexión de la derecha si el estado del parámetro booleano real correspondiente (indicado mediante xxx) está activo. De lo contrario, el estado de la conexión derecha está inactivo.
Normalmente cerrado		En el caso de los contactos normalmente cerrados, el estado de la conexión de la izquierda se transfiere a la conexión de la derecha si el estado del parámetro booleano real correspondiente (indicado mediante xxx) está inactivo. De lo contrario, el estado de la conexión derecha está inactivo.

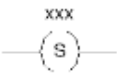

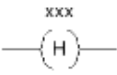
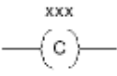
Designación	Representación	Descripción
Contacto para detectar transiciones positivas		En el caso de estos contactos, la conexión derecha de un ciclo de programa está activa si la transferencia del parámetro real correspondiente (indicado mediante xxx) pasa de inactivo a activo y el estado de la conexión izquierda está activo al mismo tiempo. De lo contrario, el estado de la conexión derecha es 0. Consulte también <i>Detección de flancos, página 401</i> .
Contacto para detectar transiciones negativas		En el caso de estos contactos, la conexión derecha de un ciclo de programa está activa si la transferencia del parámetro real correspondiente (indicado mediante xxx) pasa de activo a inactivo y el estado de la conexión izquierda está activo al mismo tiempo. De lo contrario, el estado de la conexión derecha es 0. Consulte también <i>Detección de flancos, página 401</i> .

Tipos de bobinas

Designación	Representación	Descripción
Bobina		En el caso de las bobinas, el estado de la conexión de la izquierda se transfiere al parámetro booleano real correspondiente (indicado mediante xxx) y la conexión de la derecha.
Bobina negada		En el caso de estas bobinas, el estado de la conexión de la izquierda se copia en la conexión de la derecha. El estado invertido de la conexión de la izquierda se copia al parámetro booleano real correspondiente (indicado mediante xxx). Si la conexión de la izquierda está inactiva, la conexión de la derecha también estará inactiva y el parámetro booleano real correspondiente estará activo.
Bobina para detectar transiciones positivas		En el caso de estas bobinas, el estado de la conexión de la izquierda se copia en la conexión de la derecha. El parámetro real correspondiente de tipo de dato EBOOL (indicado mediante xxx) es 1 para un ciclo de programa si se realiza una transición de la conexión de la izquierda de 0 a 1. Consulte también <i>Detección de flancos, página 401</i> .
Bobina para detectar transiciones negativas		En el caso de estas bobinas, el estado de la conexión de la izquierda se copia en la conexión de la derecha. El parámetro booleano real correspondiente (indicado mediante xxx) es 1 para un ciclo de programa si se realiza una transición de la conexión de la izquierda de 1 a 0. Consulte también <i>Detección de flancos, página 401</i> .

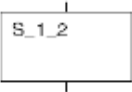
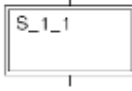
Designación	Representación	Descripción
Bloque de funcionamiento		<p>Si el estado de la conexión izquierda es 1, se ejecutará la instrucción ST incluida en el bloque. Se admiten todas las instrucciones ST (véase página 539) excepto las instrucciones de control:</p> <ul style="list-style-type: none"> • (RETURN, • JUMP, • IF, • CASE, • FOR, • etc.). <p>En el caso de los bloques de operación, el estado de la conexión izquierda se transmite a la conexión derecha independientemente del resultado de la instrucción ST. Un bloque puede contener hasta 4.096 caracteres. Si no se pueden representar todos los caracteres, se representa el inicio de la secuencia de caracteres seguido de puntos suspensivos (...). Un bloque de operación ocupa 1 fila y 4 columnas. Ejemplo:</p>  <p>En el ejemplo, Instrucción1 se ejecuta si Entrada1=1. Instrucción2 se ejecuta si Entrada1=1 y Entrada2=1 (el resultado de Instrucción1 no influye en la ejecución de Instrucción2). Salida1 será 1 cuando Entrada1 = 1 y Entrada2 = 1 (los resultados de Instrucción1 e Instrucción2 no influyen en el estado de Salida1).</p>

Designación	Representación	Descripción
<p>Bloque de comparación horizontal</p>		<p>Los bloques de comparación horizontal sirven para ejecutar una expresión de comparación (<, >, <=, >=, =, <>) en el lenguaje de programación ST. (Nota: Se puede conseguir la misma funcionalidad mediante las expresiones ST (véase página 539)). El bloque de comparación realiza un AND de su pin de entrada de la izquierda y el resultado de su condición de comparación asigna el resultado de este AND a su pin de salida incondicionalmente.</p> <p>Por ejemplo, si el estado de la conexión izquierda es 1 y el resultado de la comparación es 1, el estado de la conexión derecha será 1.</p> <p>Un bloque de comparación horizontal puede contener hasta 4.096 caracteres. Si no se pueden representar todos los caracteres, se representa el inicio de la secuencia de caracteres seguido de puntos suspensivos (...).</p> <p>Un bloque de comparación horizontal ocupa una fila y dos columnas.</p> <p>Ejemplo:</p>  <p>En el ejemplo, Comparación1 se ejecuta si Entrada1=1. Comparación2 se ejecuta si Entrada1=1, Entrada2=1, el resultado de Comparación1=1. Salida1 es 1 si Entrada1=1, Entrada2=1, el resultado de Comparación1=1 y el resultado de Comparación2=1.</p>

Designación	Representación	Descripción
Bobina de ajuste		En el caso de estas bobinas, el estado de la conexión de la izquierda se copia en la conexión de la derecha. El parámetro booleano real correspondiente (indicado mediante xxx) está activo si la conexión de la izquierda está activa. Si no, permanece sin cambios. El parámetro booleano real correspondiente puede restablecerse mediante la bobina de ajuste. Consulte también <i>Detección de flancos, página 401</i> .
Bobina de restablecimiento		En el caso de estas bobinas, el estado de la conexión de la izquierda se copia en la conexión de la derecha. El parámetro booleano real correspondiente (indicado mediante xxx) está inactivo si la conexión de la izquierda está activa. Si no, permanece sin cambios. El parámetro booleano real correspondiente puede establecerse mediante la bobina de ajuste. Consulte también <i>Detección de flancos, página 401</i> .
Bobina de detención		En el caso de estas bobinas, si el estado de la conexión de la izquierda es 1, la ejecución del programa se detiene de forma inmediata. (En las bobinas de detención, el estado de la conexión de la izquierda no se copia en la conexión de la derecha.)
Bobina de llamada		En el caso de estas bobinas, el estado de la conexión de la izquierda se copia en la conexión de la derecha. Si el estado de la conexión de la izquierda está activo, se produce una llamada al subprograma correspondiente (indicado mediante xxx). La subrutina a la que se va a llamar debe encontrarse en la misma tarea que la sección LD invocante. También es posible llamar subrutinas ubicadas dentro de otras subrutinas. Las subrutinas son una ampliación de la norma CEI 61131-3 y se deben habilitar de forma explícita. En las secciones de acción SFC, sólo se admiten bobinas de llamada (llamadas de subrutina) si está habilitada la modalidad Multi-Token.

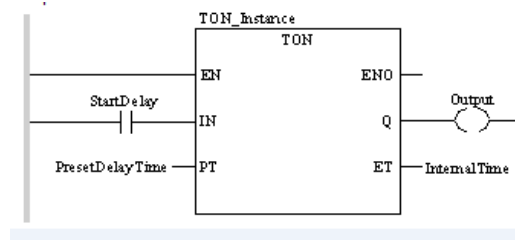
Elementos SFC

Tipos de pasos

Tipo	Representación	Descripción
Paso "normal"		Un paso se activa cuando el paso anterior se convierte en inactivo (debe transcurrir un tiempo de retardo definido) y se cumple la transición preconectada. Normalmente, un paso se convierte en inactivo cuando transcurre el tiempo de retardo definido y se cumple la condición postconectada. En el caso de las conjunciones simultáneas, todos los pasos anteriores deben cumplir estas condiciones. A cada paso le corresponden cero o más acciones. Los pasos sin acción se denominan pasos de espera.
Paso inicial		El estado inicial de una cadena de ejecución está caracterizado por el paso inicial. Tras la inicialización del proyecto o de la cadena de ejecución, el paso inicial está activo. A los pasos iniciales no se les suele asignar ninguna acción. En Single-Token (conforme a IEC 61131-3), sólo se admite un único paso inicial por secuencia. En Multi-Token se admite una cantidad definible de pasos iniciales (de 0 a 100).

Temporizador

El TON es el bloque de función que se utiliza como retardo de conexión. El estado inicial de ET durante la primera llamada del bloque de función es 0. Como parámetros adicionales se pueden configurar EN y ENO.



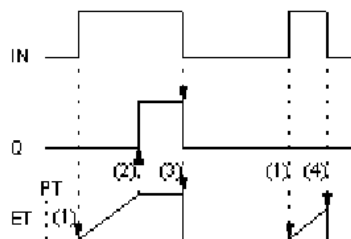
Sus parámetros de entrada son:

Parámetro	Tipo de datos	Significado
IN	BOOL	Iniciar retardo
PT	TIME	Ajuste previo del tiempo de retardo

Sus parámetros de salida son:

Parámetro	Tipo de datos	Significado
Q	BOOL	Salida
ET	TIME	Tiempo interno

Representación del retardo de conexión TON:



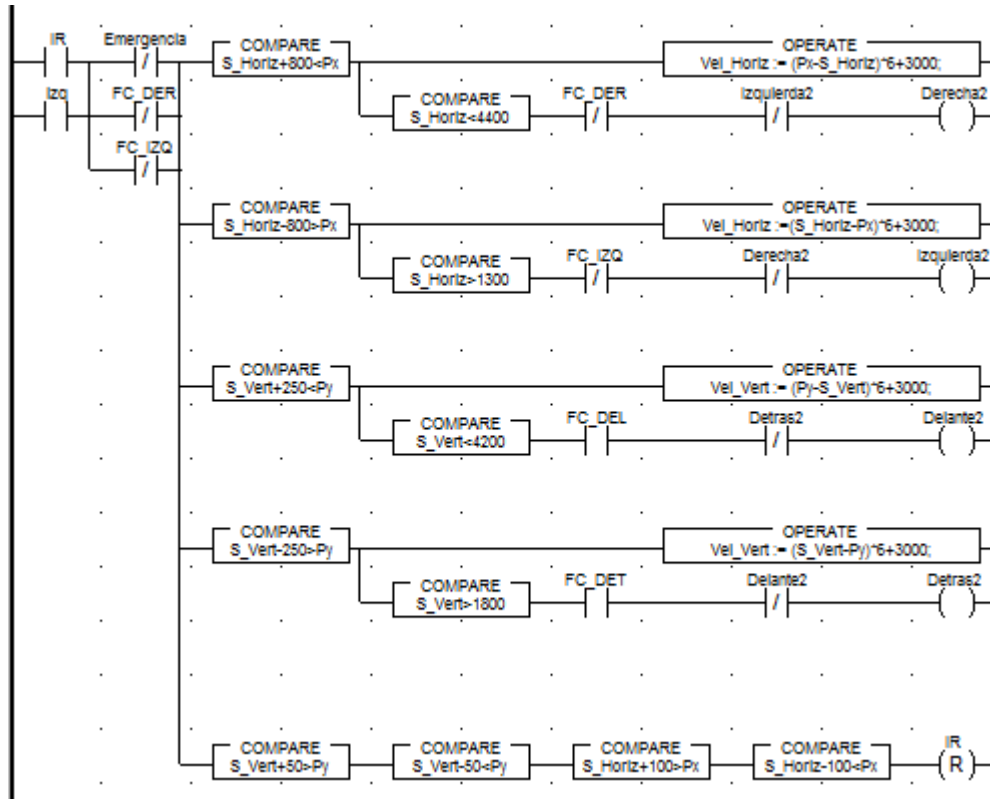
- (1) Si IN se convierte en 1, se inicia el tiempo interno (ET).
- (2) Si el tiempo interno alcanza el valor de PT, Q se convierte en 1.
- (3) Si IN se convierte en 0, Q se convierte en 0 y se detiene/restablece el tiempo interno.
- (4) Si IN se convierte en 0 antes de que el tiempo interno haya alcanzado el valor de PT, se detiene/ restablece el tiempo interno, sin que Q se haya convertido en 1.

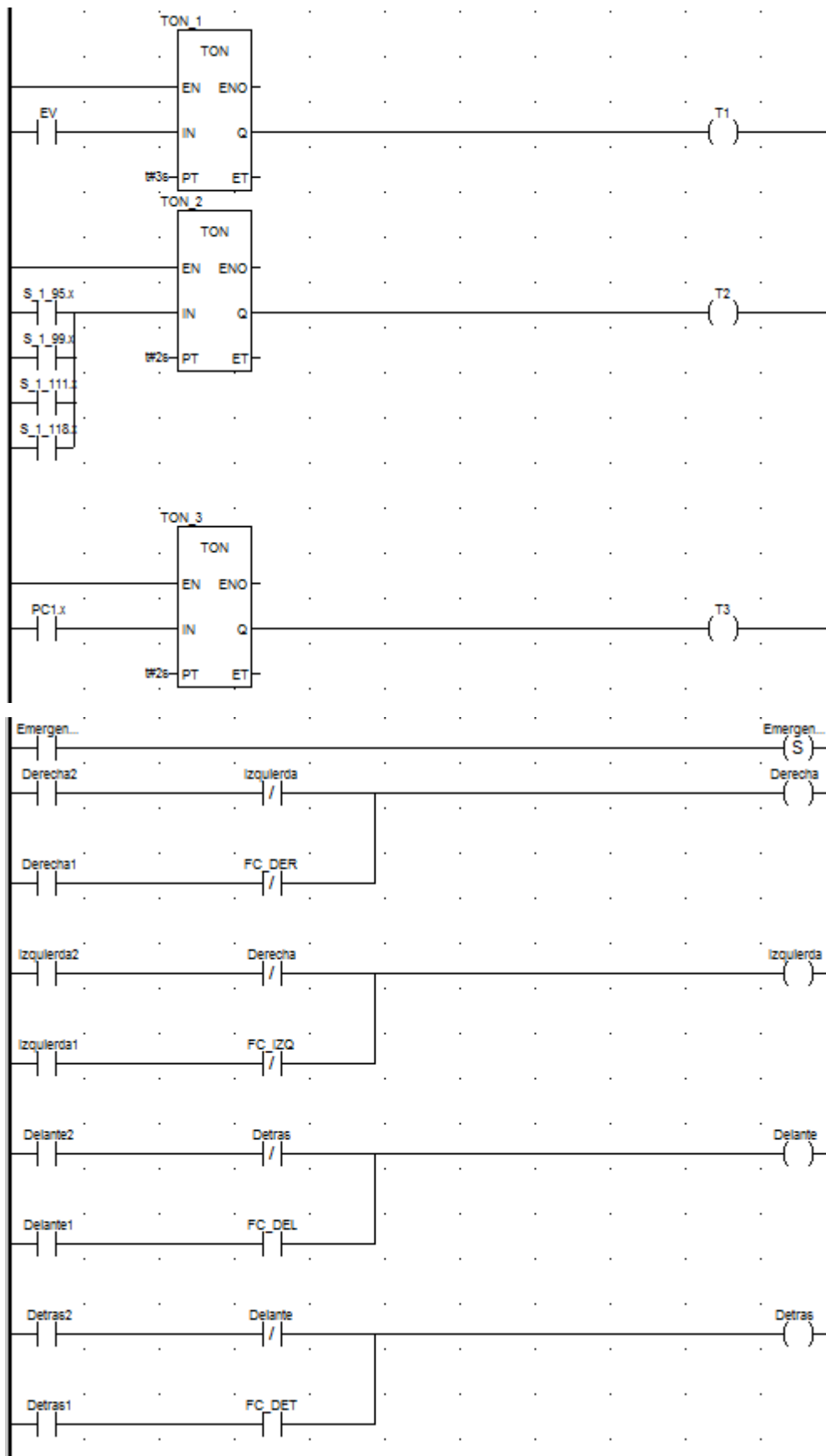
8.2 Parámetros utilizados

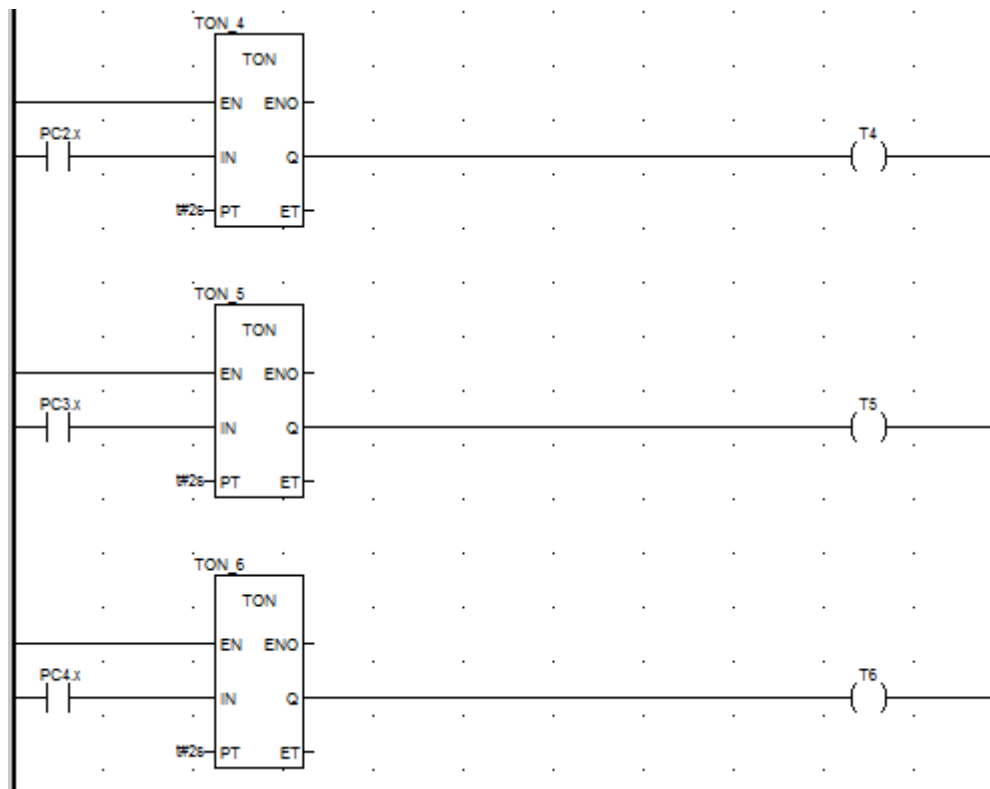
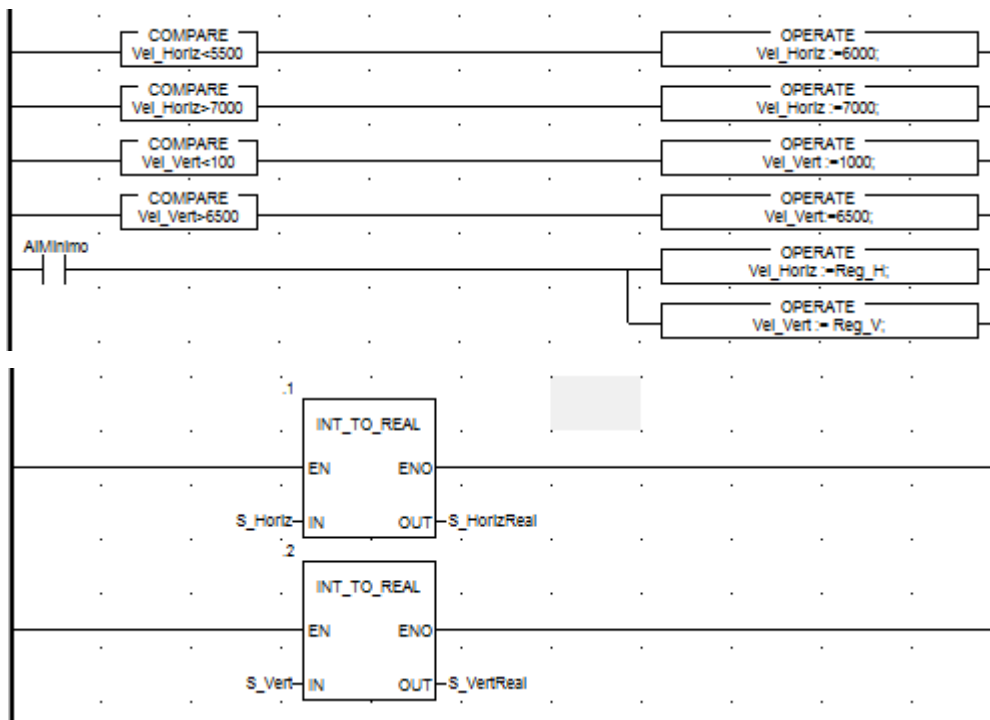
Variable	Tipo	Dirección	Función
Delante	EBOOL	%Q.0.1.21	Entrada del variador vertical indicadora del sentido
Derecha	EBOOL	%Q.0.1.6	Entrada del variador horizontal indicadora del sentido
Detrás	EBOOL	%Q.0.1.20	Entrada del variador vertical indicadora del sentido
Izquierda	EBOOL	%Q.0.1.17	Entrada del variador horizontal indicadora del sentido
EV	EBOOL	%Q.0.3.18	Control de la electro-válvula
FC_DEL	EBOOL	%I.0.3.1	Final de carrera delante
FC_DER	EBOOL	%I.0.3.2	Final de carrera derecho
FC_DET	EBOOL	%I.0.3.3	Final de carrera detrás
FC_IZQ	EBOOL	%I.0.3.4	Final de carrera izquierda
S_HORIZ	INT	%IW.0.2.3	Entrada analógica del sensor horizontal que indica la posición
S_VERT	INT	%IW.0.2.2	Entrada analógica del sensor vertical que indica la posición
Taladro	EBOOL	%Q.0.2.20	Control del contactor que activa el taladro
VEL_Hor_1	EBOOL	%Q.0.2.18	Entrada del variador horizontal que indica velocidad
VEL_Hor_2	EBOOL	%Q.0.2.19	Entrada del variador horizontal que indica velocidad
VEL_Horiz	INT	%QW.0.2.4	Entrada analógica del variador horizontal
VEL_Vert	INT	%QW.0.2.5	Entrada analógica del variador vertical
VEL_Vert_1	EBOOL	%Q.0.2.18	Entrada del variador vertical que indica velocidad
VEL_Vert_2	EBOOL	%Q.0.2.19	Entrada del variador vertical que indica velocidad

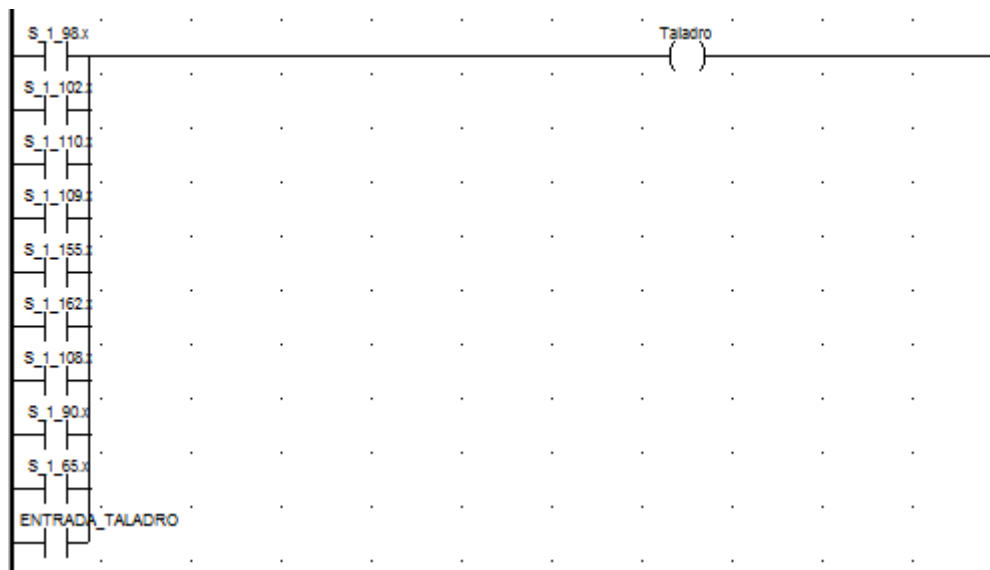
8.3. Programa

Esquema LD Movimiento

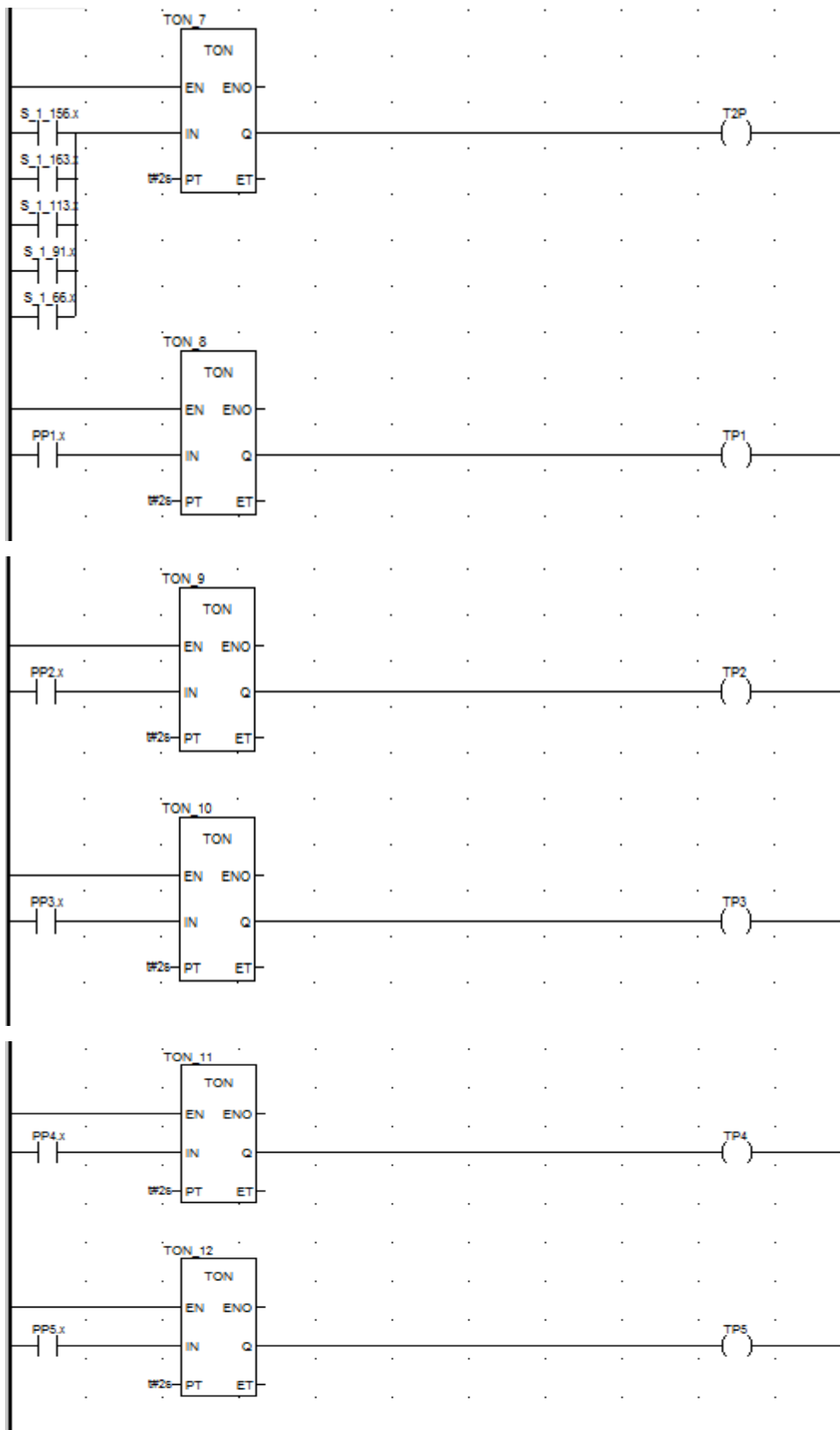




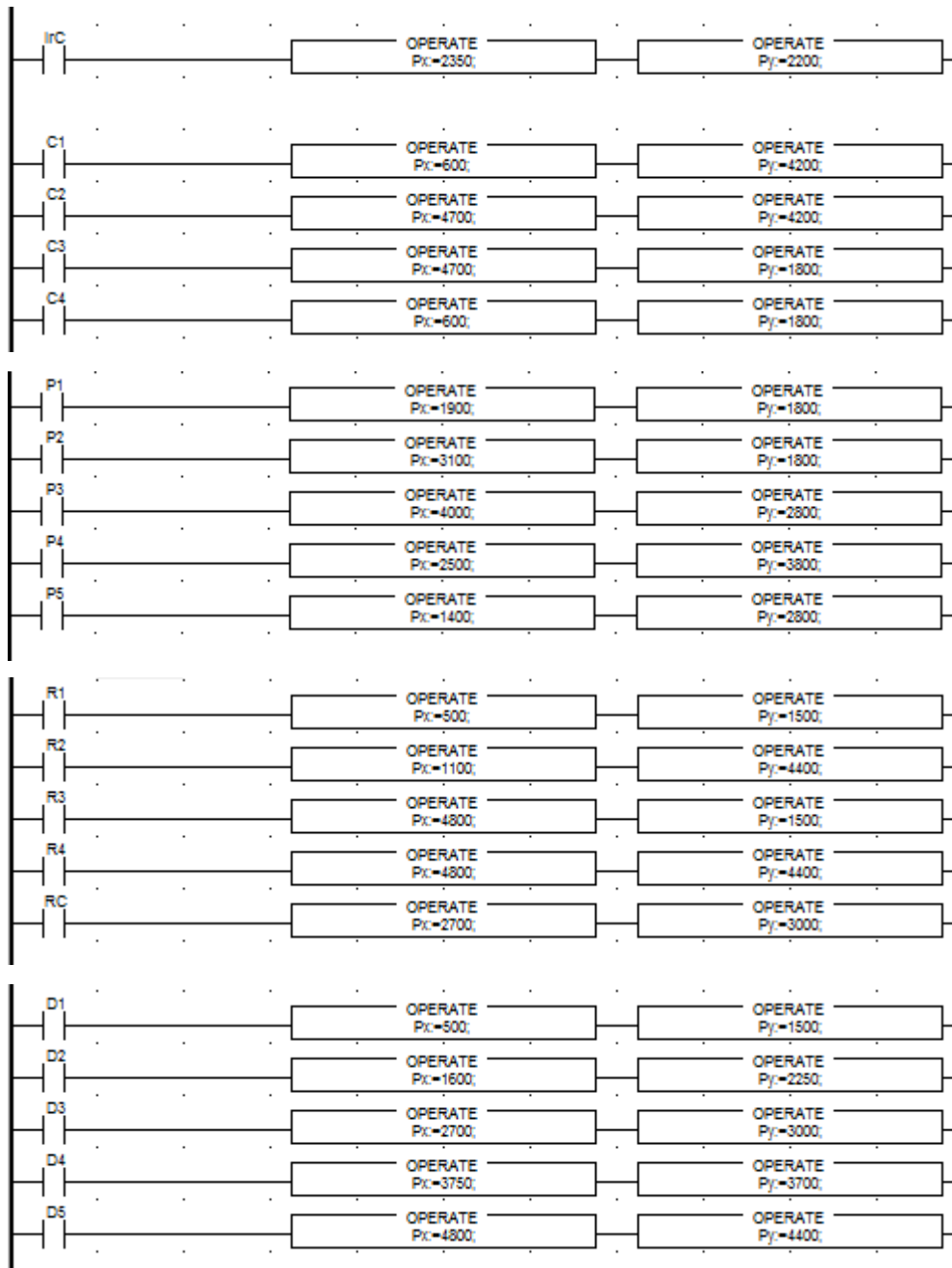




Esquema LD PENTAGRAMA

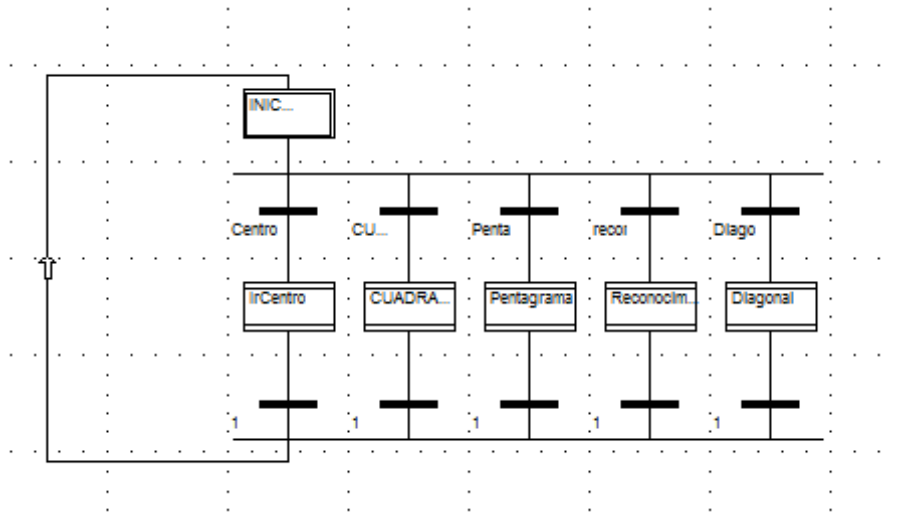


Esquema LD Puntos



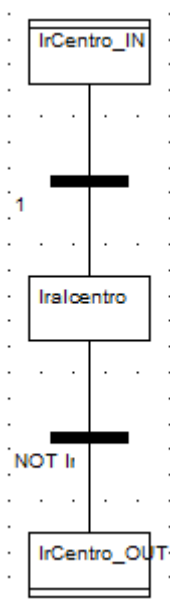
Esquema SFC

Esquema primario

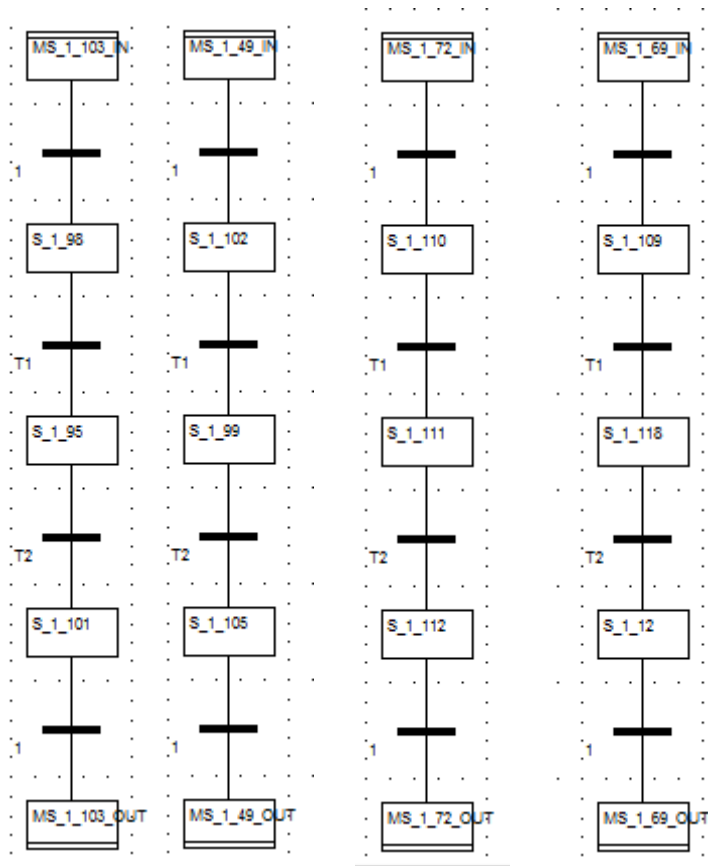
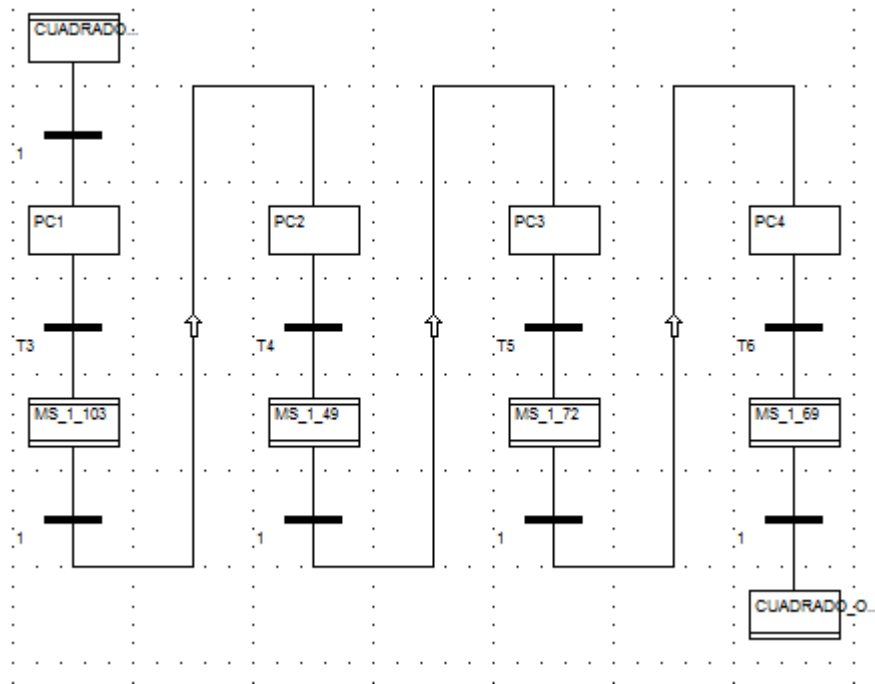


Esquemas secundarios

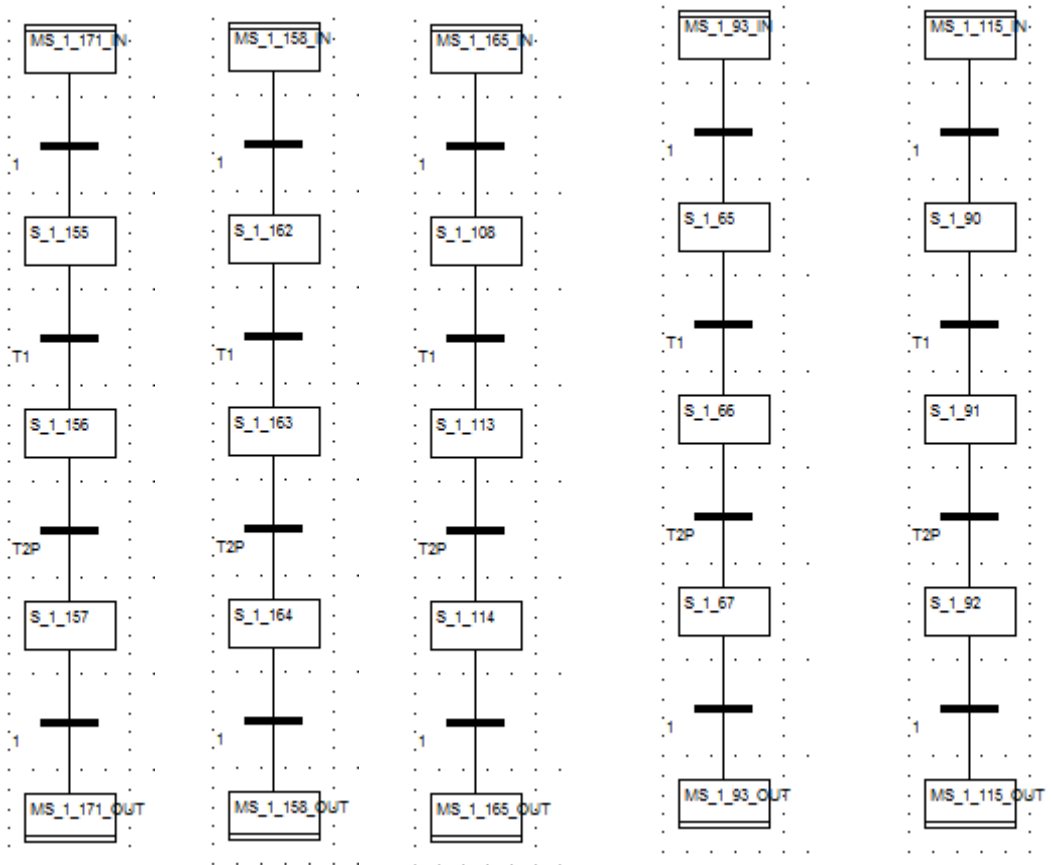
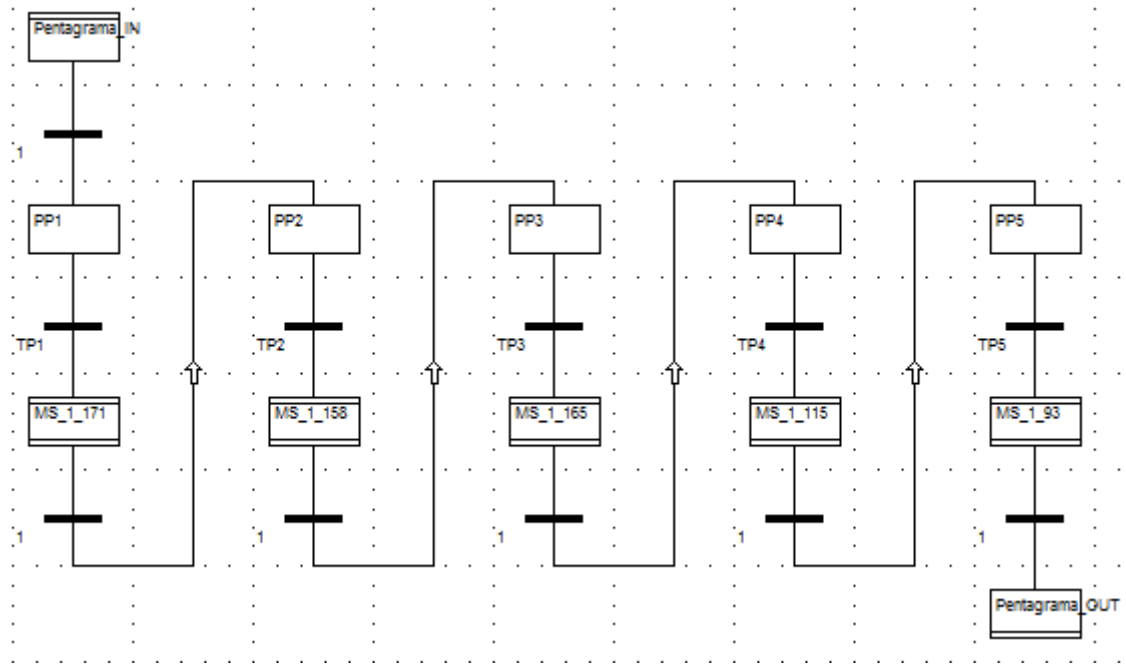
IrCentro



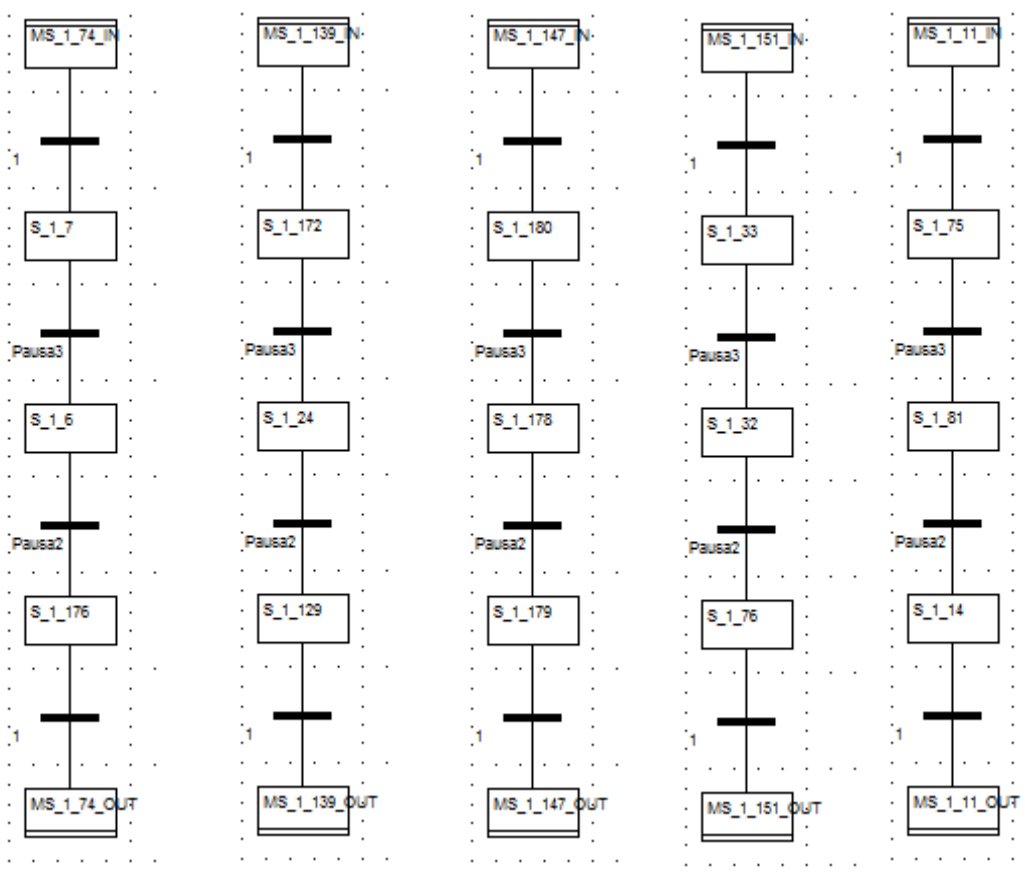
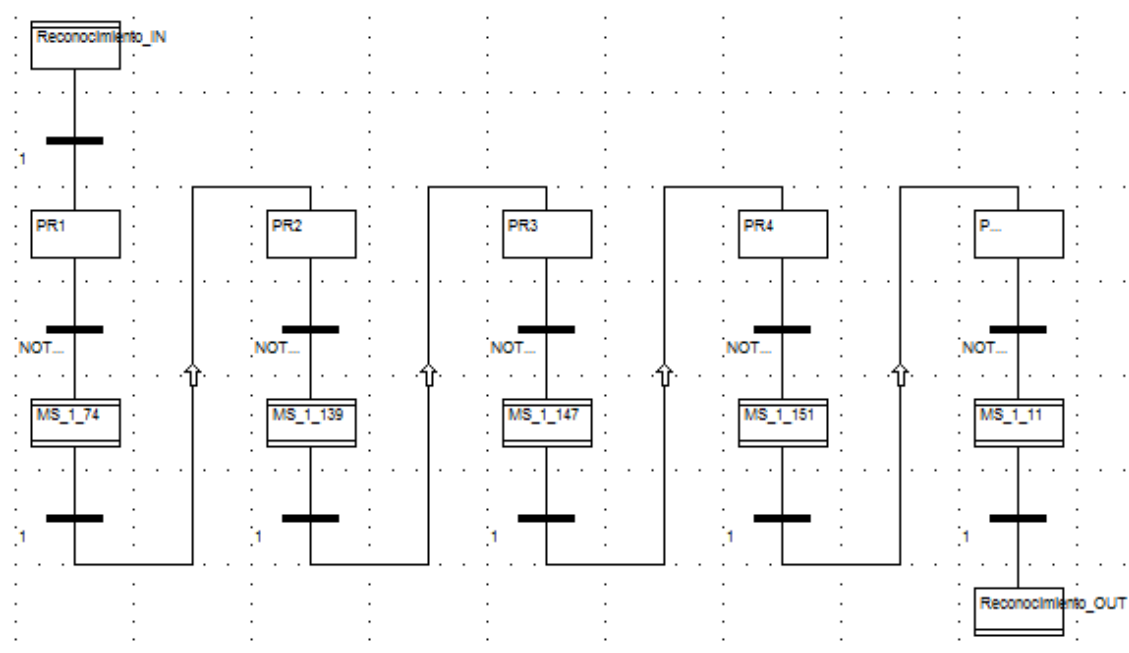
Cuadrado



Pentagrama



Reconocimiento



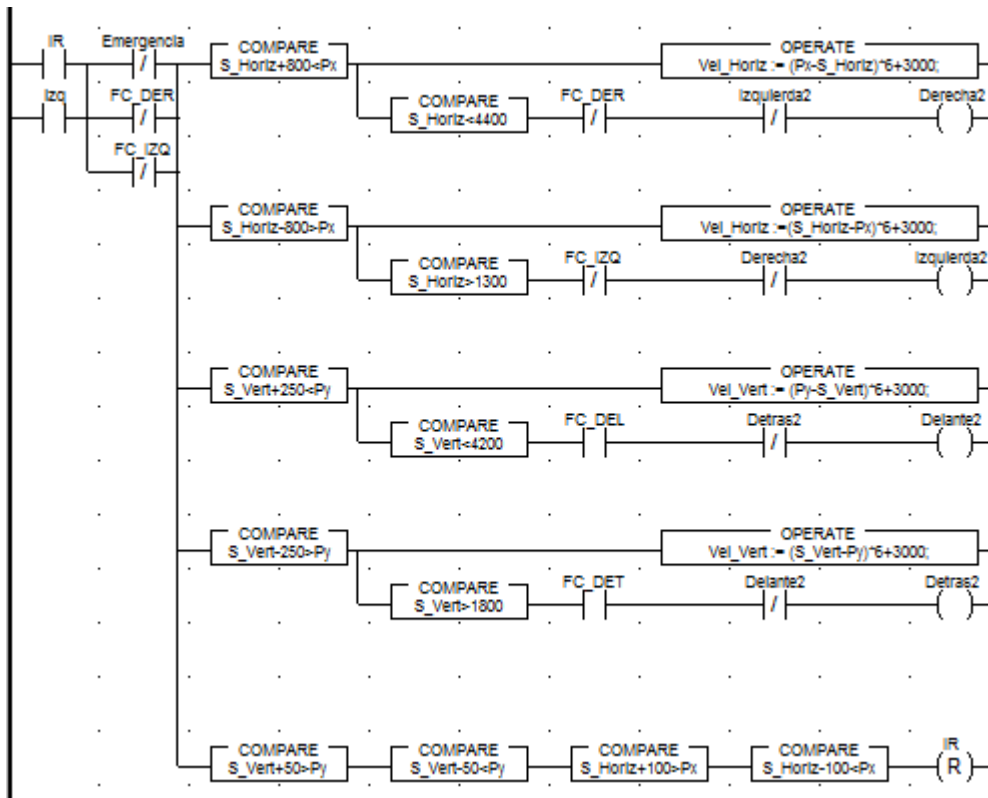
8.4. Explicación del programa

En las siguientes páginas se va a explicar las diferentes partes del programa, explicando la función que realizan, como actúan y las variables que utiliza.

8.4.1 Movimiento

Esta parte del programa es de crucial importancia, debido a que permite el movimiento de la máquina como su propio nombre indica.

Vamos a explicar las acciones que realiza línea por línea.



Al principio vemos el contacto cerrado IR, éste permite que podamos pasar a las siguientes 5 líneas que vienen a continuación:

1. En la primera línea se encuentra un bloque Compare que compara el valor de S_Horiz, que es la variable que modifica su valor con el sensor lineal y así determina donde está la máquina en el eje horizontal dependiendo del valor de Px, el valor que controlamos nosotros para

indicar que se mueva la máquina por el eje x. Si se cumple la condición del primer compare, entonces pasamos a las siguientes dos líneas. Primero el Operate marcará el valor de la velocidad con Vel_Horiz dependiendo de la distancia que deba recorrer la máquina hasta la posición indicada inyectará más o menos potencia a los motores.

Segundo, el bloque Compare es el indicador de la posición nada más empezar el movimiento, es el encargado de averiguar si la máquina debe de ir hacia la derecha o no, también dispone de unos contactos cerrados, el de FC_DER que será el encargado de parar el movimiento hacia la derecha y el contacto de Izquierda2 encargado de desconectar Izquierda2 y así no crear ningún conflicto entre el movimiento derecha/izquierda.

2. En la segunda línea se encuentra un bloque Compare que compara el valor de S_Horiz, que es la variable que modifica su valor con el sensor lineal y así determina donde está la máquina en el eje horizontal dependiendo del valor de Px, el valor que controlamos nosotros para indicar que se mueva la máquina por el eje x. Si se cumple la condición del primer compare, entonces pasamos a las siguientes dos líneas. Primero el Operate marcará el valor de la velocidad con Vel_Horiz dependiendo de la distancia que deba recorrer la máquina hasta la posición indicada inyectará más o menos potencia a los motores. Segundo, el bloque Compare es el indicador de la posición nada más empezar el movimiento, es el encargado de averiguar si la máquina debe de ir hacia la derecha o no, también dispone de unos contactos cerrados, el de FC_IZQ que será el encargado de parar el movimiento hacia la izquierda y el contacto de derecha2 encargado de desconectar derecha2 y así no crear ningún conflicto entre el movimiento derecha/izquierda.
3. En la tercera línea se encuentra un bloque Compare que compara el valor de S_Vert, que es la variable que modifica su valor con el sensor

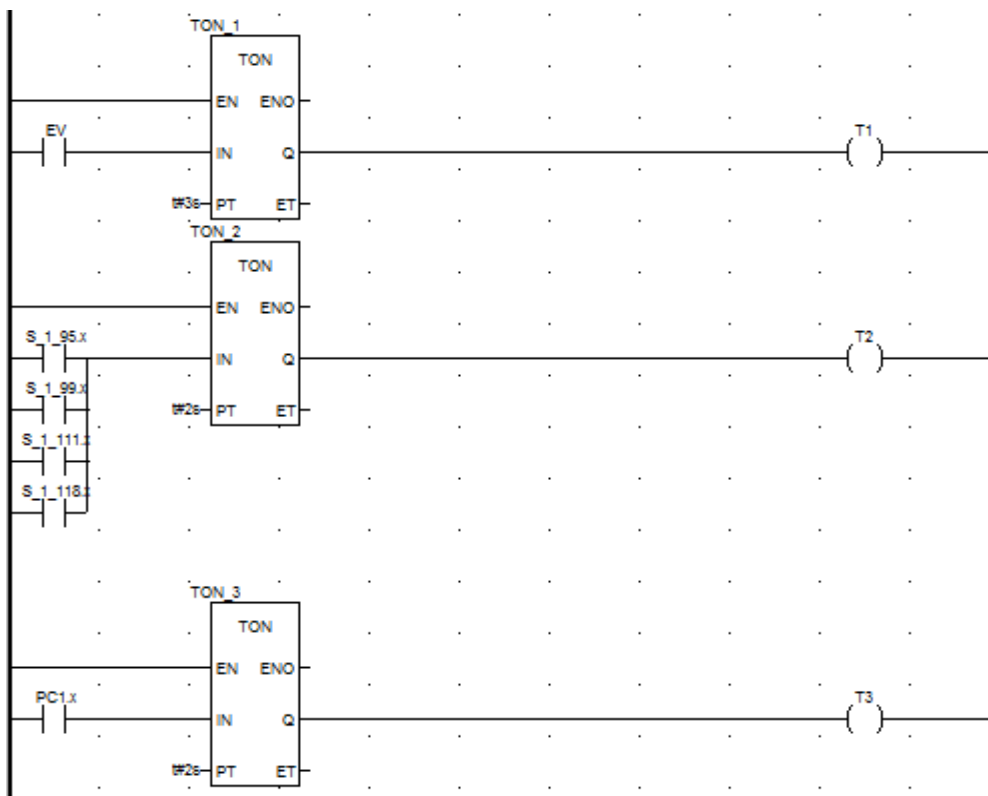
lineal y así determina donde está la máquina en el eje vertical dependiendo del valor de Py, el valor que controlamos nosotros para indicar que se mueva la máquina por el eje y. Si se cumple la condición del primer compare, entonces pasamos a las siguientes dos líneas. Primero el Operate marcará el valor de la velocidad con Vel_Vert dependiendo de la distancia que deba recorrer la máquina hasta la posición indicada inyectará más o menos potencia a los motores.

Segundo, el bloque Compare es el indicador de la posición nada más empezar el movimiento, es el encargado de averiguar si la máquina debe de ir hacia la delante o no, también dispone de unos contactos cerrados, el de FC_DEL que será el encargado de parar el movimiento hacia la delante y el contacto de Detrás2 encargado de desconectar Detrás2 y así no crear ningún conflicto entre el movimiento delante/detrás.

4. En la cuarta línea se encuentra un bloque Compare que compara el valor de S_Vert, que es la variable que modifica su valor con el sensor lineal y así determina donde está la máquina en el eje vertical dependiendo del valor de Py, el valor que controlamos nosotros para indicar que se mueva la máquina por el eje y. Si se cumple la condición del primer compare, entonces pasamos a las siguientes dos líneas. Primero el Operate marcará el valor de la velocidad con Vel_Vert dependiendo de la distancia que deba recorrer la máquina hasta la posición indicada inyectará más o menos potencia a los motores. Segundo, el bloque Compare es el indicador de la posición nada más empezar el movimiento, es el encargado de averiguar si la máquina debe de ir hacia la detrás o no, también dispone de unos contactos cerrados, el de FC_DET que será el encargado de parar el movimiento hacia la detrás y el contacto de Delante2 encargado de desconectar Delante2 y así no crear ningún conflicto entre el movimiento delante/detrás.

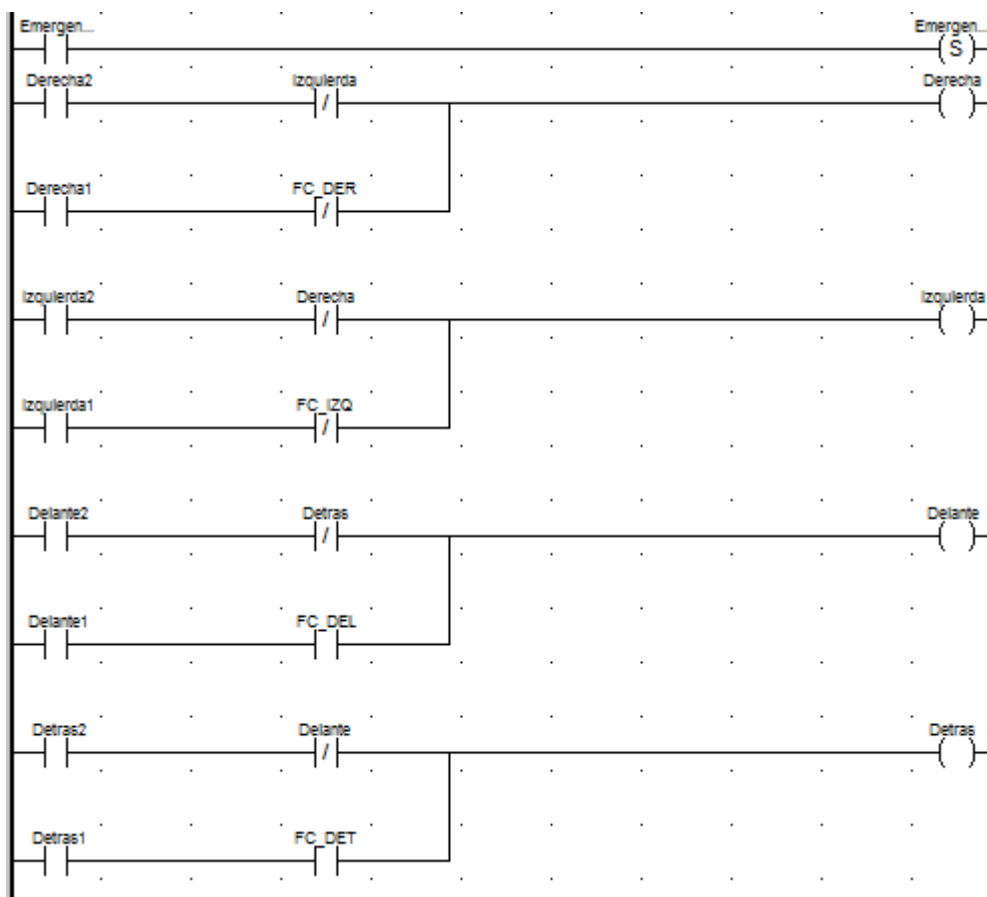
5. La quinta línea se activa cuando hemos llegado ya al punto de destino y reinicia IR para abrir el contacto IR y así cortar cualquier acción, por desgracia este se ha debido crear con un pequeño margen de error debido a que la velocidad mínima necesitada por el motor para moverse es elevada, lo que concluye sino en que la máquina se salta el punto indicado

La variable izq se ha creado para permitir el uso de la máquina de forma completamente manual, es decir, se ha creado una pantalla con diferentes botones para poder controlar cada movimiento de forma totalmente manual con la pantalla táctil Magelis, a través de webgate o incluso con nuestro dispositivo android.

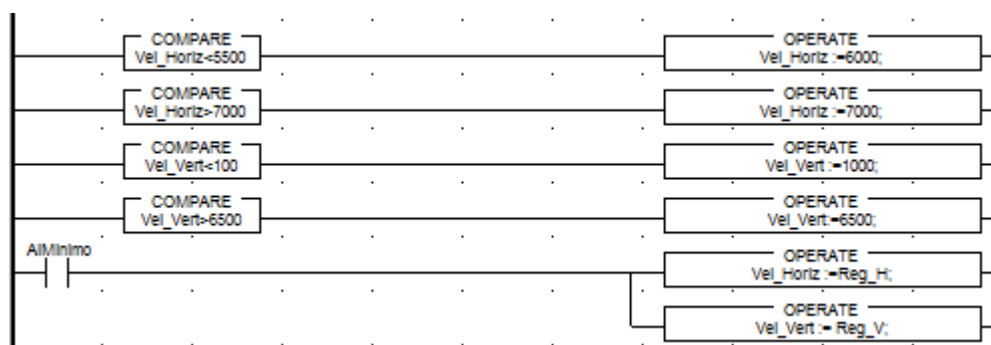


Esta parte del programa es la encargada del movimiento de taladrar coordinado con la electroválvula, es decir, cuando llegamos al punto PC1 se activa la electroválvula para que el taladro inicie el descenso

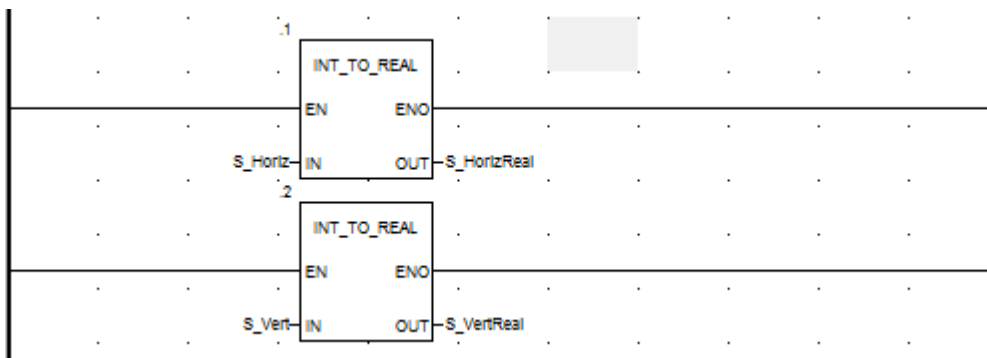
hacia el primer punto del cuadrado marcado, este es un paso que se repite en cada esquina del cuadrado, de ahí los 4 contactos que pueden controlar la activación de la transición T2



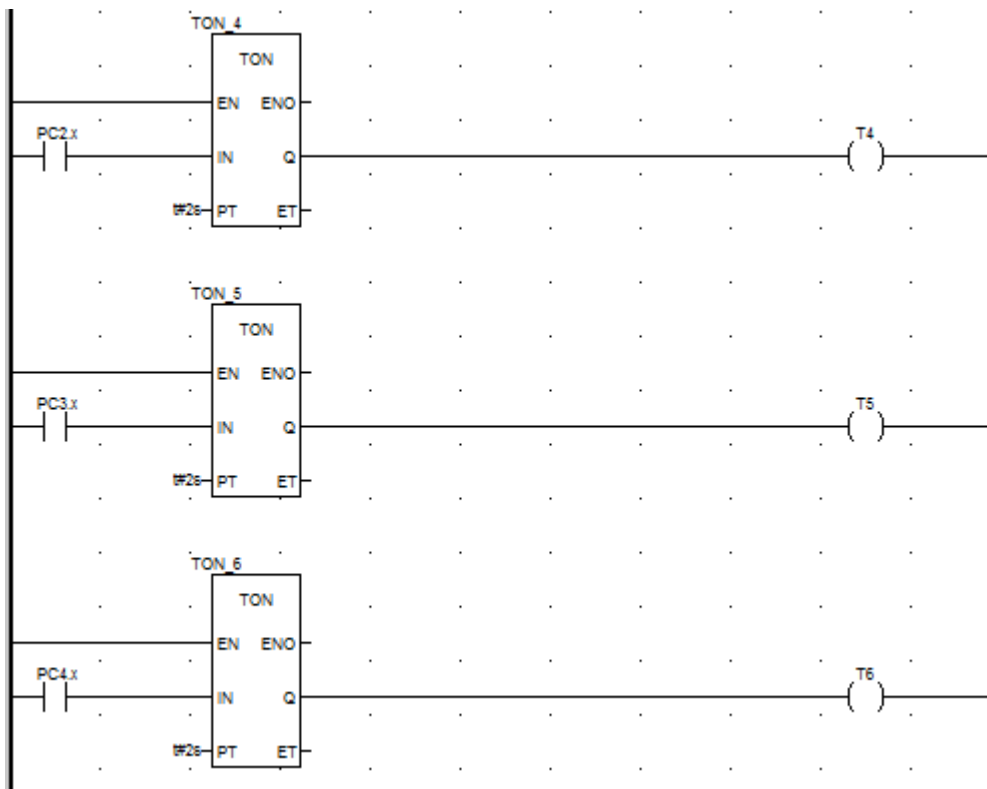
Esta parte se ha hecho por seguridad, está nunca es suficiente, entonces para no cometer un error de solapamiento de señales hemos dividido los comandos de los movimientos del modo manual y automático.



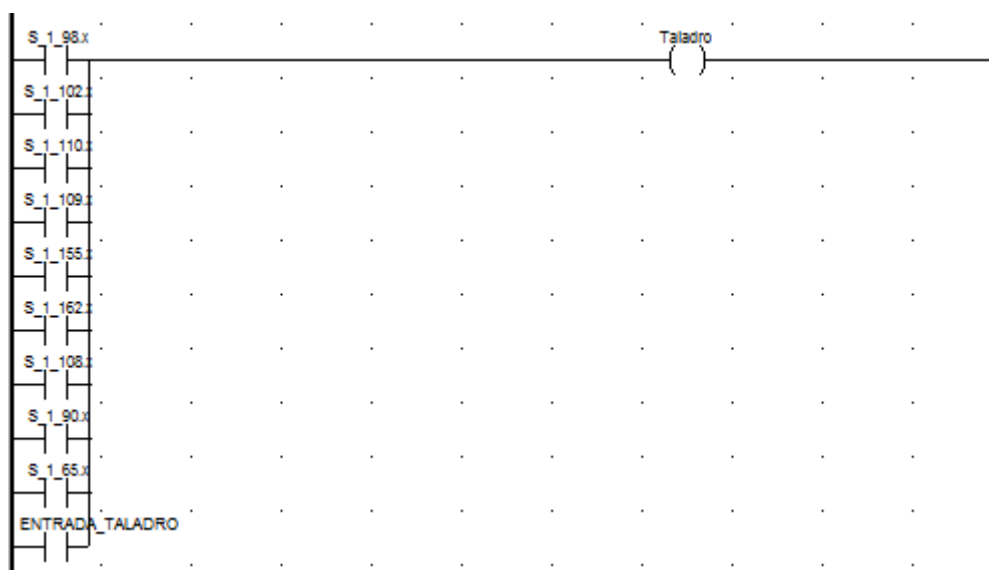
Este pedazo del programa permite poner unos rangos máximos y mínimos de la velocidad del motor. Esto lo hemos hecho debido que el movimiento mecánico necesita suficiente potencia para mover toda la estructura, si insertamos una velocidad menor, la máquina no se podrá mover, al igual que si le damos demasiada potencia, la máquina no será manejable ya que se pasa los puntos de las coordenadas marcadas. La última línea se ha diseñado para poder controlar la velocidad manualmente activando la variable AIMinimo.



Estos bloques permiten cambiar una variable INT a un a variable REAL, esto simplemente se ha hecho debido que los marcadores numéricos del software solo permiten visualizar una variable REAL a tiempo real.

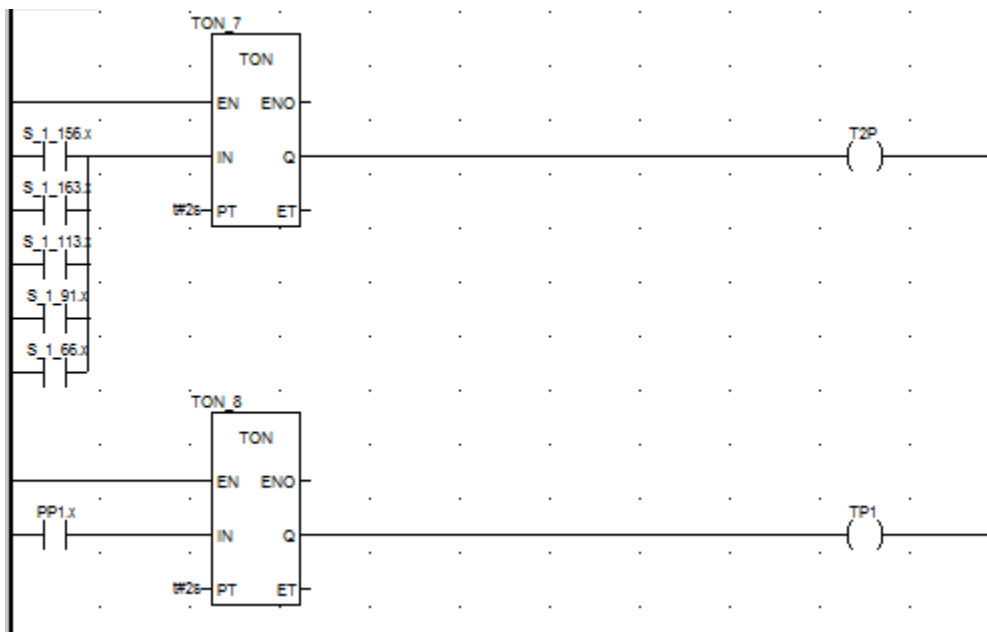


Esos temporizadores se han colocado para controlar el tiempo entre los diferentes puntos marcados a taladrar. Esto se ha realizado de esta manera para que la máquina se pueda adaptar a cualquier superficie a taladrar, ya que una superficie dura necesitará más tiempo de perforación que una superficie más blanda.

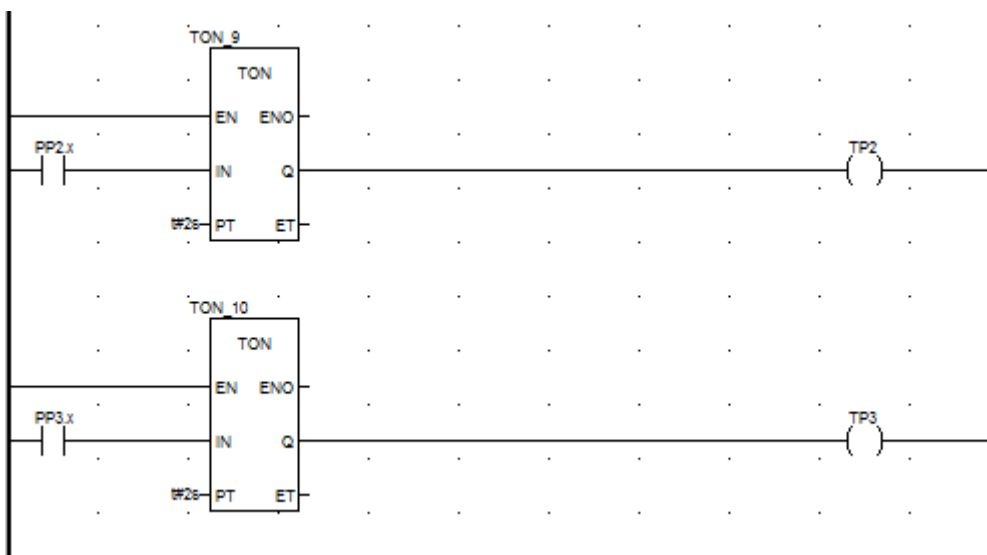


Aquí vemos todos los contactos del programa Cuadrado, Pentágono y también el botón manual que activan el taladro, también adaptable al programa que queramos realizar con la máquina, ya que se podrían insertar todas las figuras que se pueda imaginar.

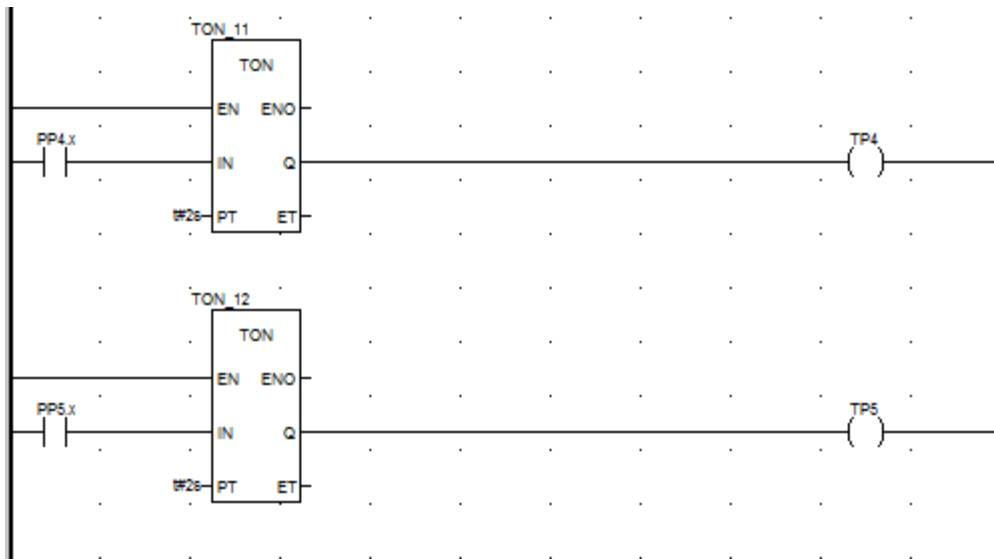
8.4.2. Pentagrama



Esta parte del programa se ha diseñado como extensión del programa de movimiento, ya que no quedaba espacio para más elementos, de nuevo tenemos los temporizadores que marcan el tiempo que debe de estar el taladro perforando en cada punto, hasta pasar al siguiente.

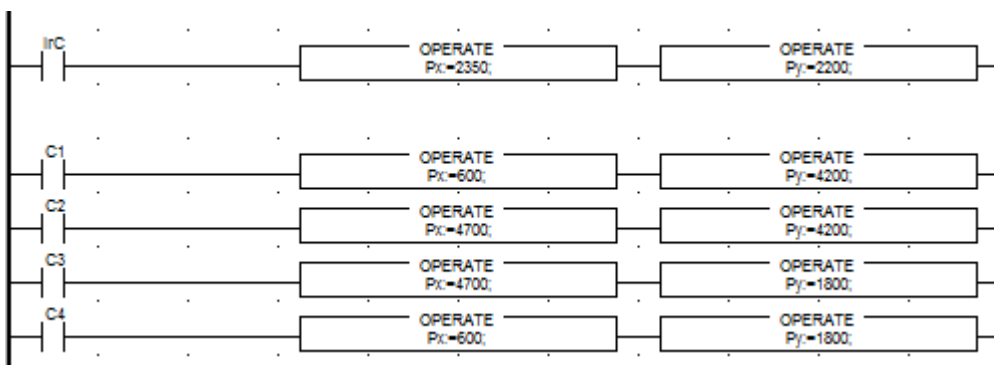


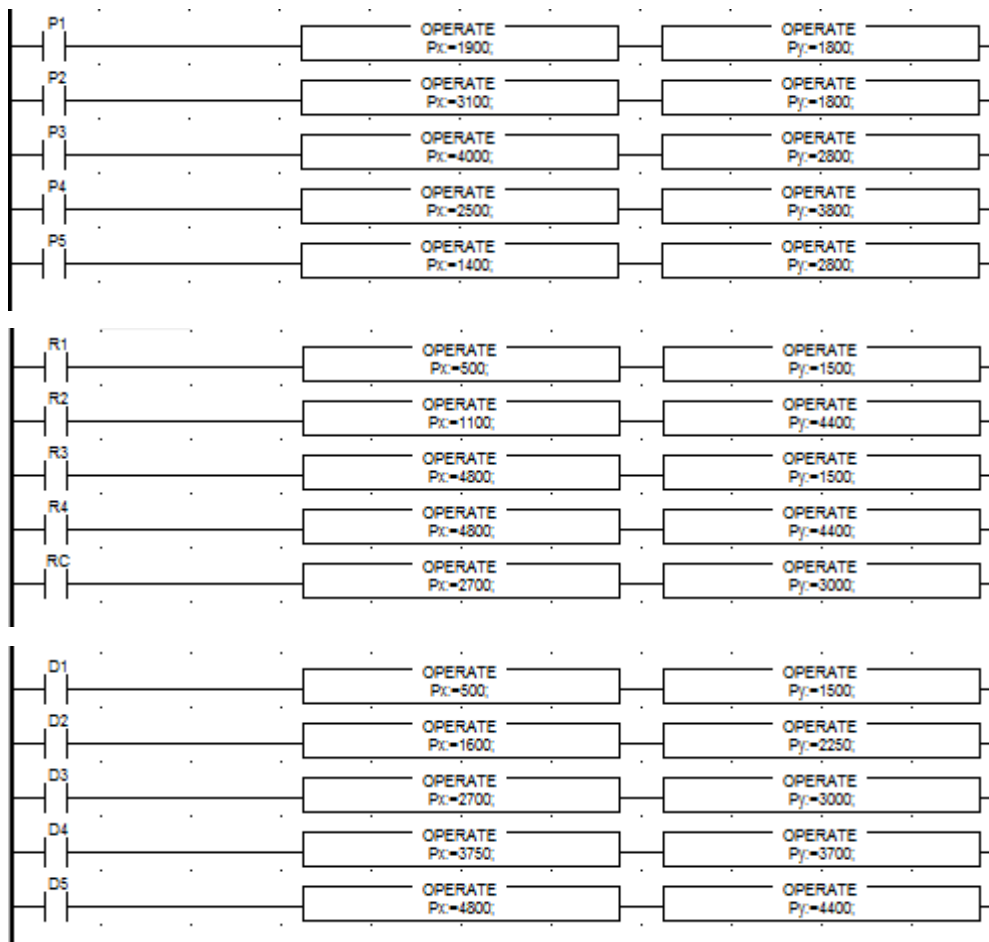
Estos dos temporizadores manejan el cambio del punto 2 del pentágono al punto 3.



Estos dos temporizadores manejan el cambio del punto 4 del pentágono al punto 5.

8.4.3. Esquema LD Puntos





En este apartado del programa están todos los puntos programados en coordenadas x e y, con esto podemos modificar el tamaño de las figuras grabadas en el programa.

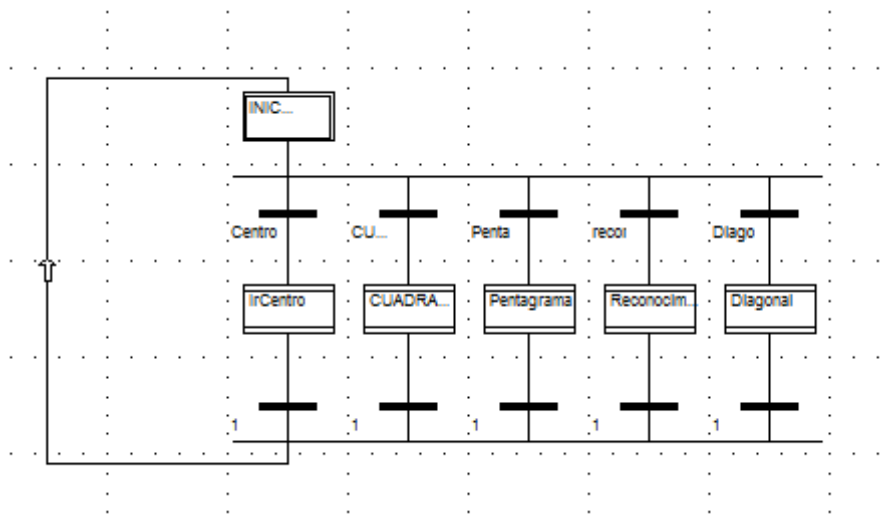
8.4.4. Esquema SFC primario

El esquema primario es el encargado de manejar el orden de todos los programas SFC secundarios.

El bit empieza en el bloque inicial, es decir, es donde arrancamos al poner en Run el programa, y a partir de ahí dependiendo del botón que apretemos entraremos en los siguientes programas comentados al principio:

- Centro
- Cuadrado
- Pentágono
- Reconocimiento
- Diagonal

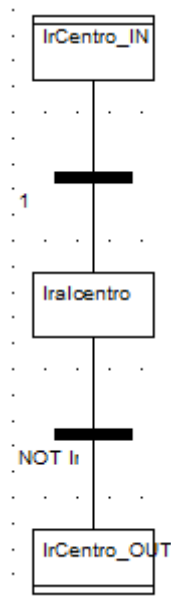
Una vez recorrido el proceso, el bit volverá al bloque inicial.



8.4.5. Esquemas SFC secundarios

El esquema secundario es el que realiza el programa según nuestra elección, a continuación explicaremos cada uno de ellos

8.4.5.1. Ir al centro

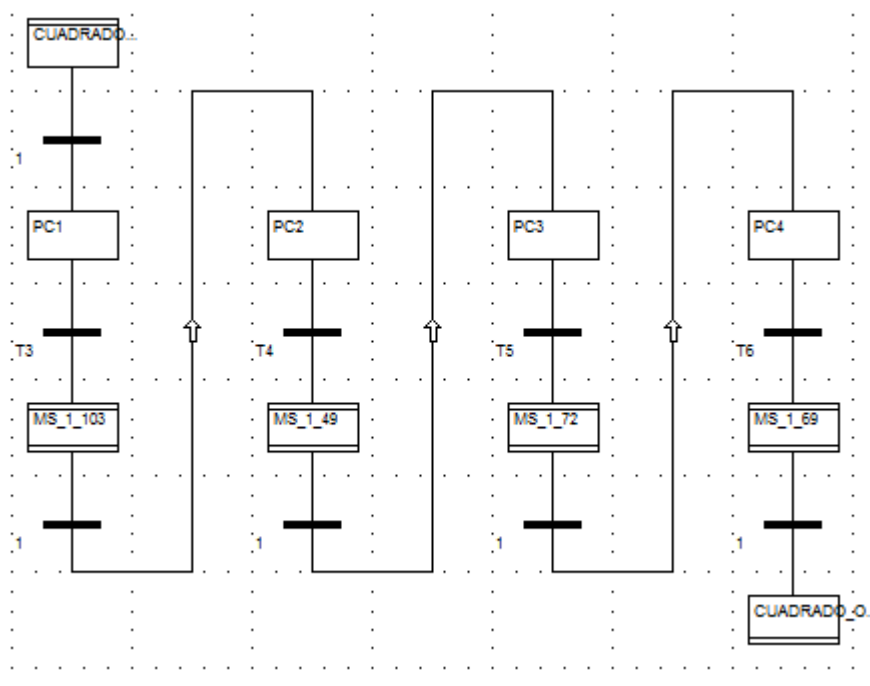


Este programa le manda al taladro colocarse en el centro de su zona de trabajo, según los puntos que se le hayan marcado como centro al principio.

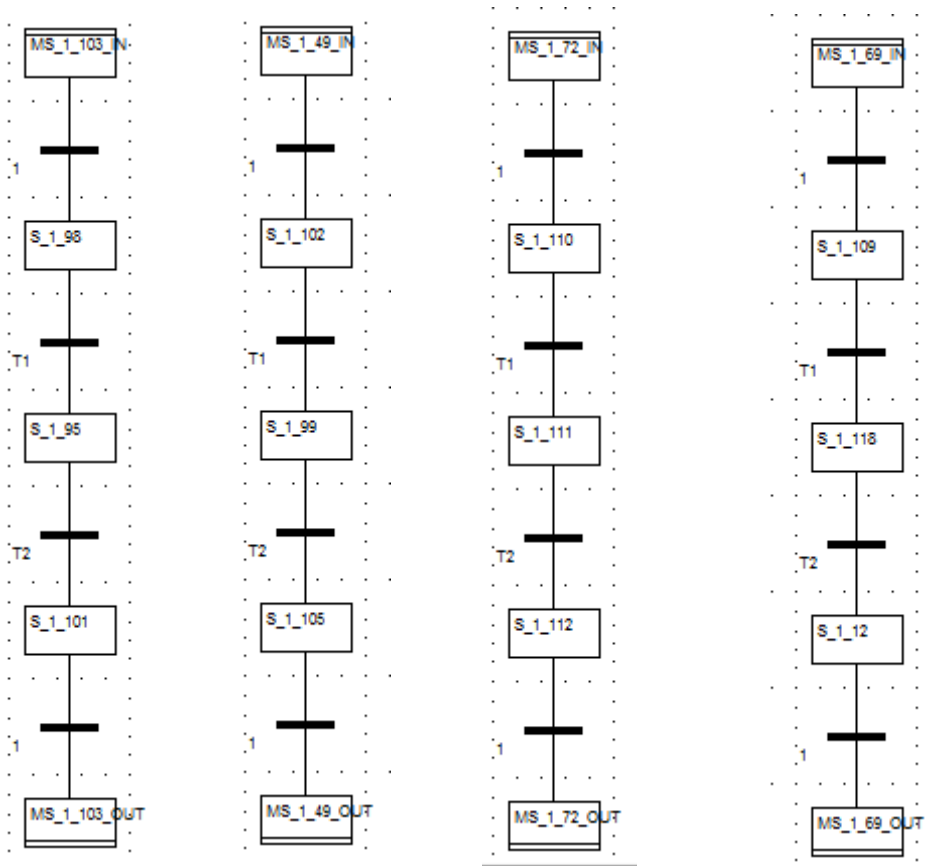
Al principio tenemos el bloque de entrada al programa, al cual habremos entrado apretando el botón Centro, a partir de aquí cruzará la primera

transición ya que le hemos puesto un 1, esto significa que el bit tiene vía libre para pasar al siguiente bloque, que activará las variables IrC e IR para activar los movimientos necesarios para colocarse la máquina en el centro, según los valores Px y Py asignados.

8.4.5.2. Cuadrado

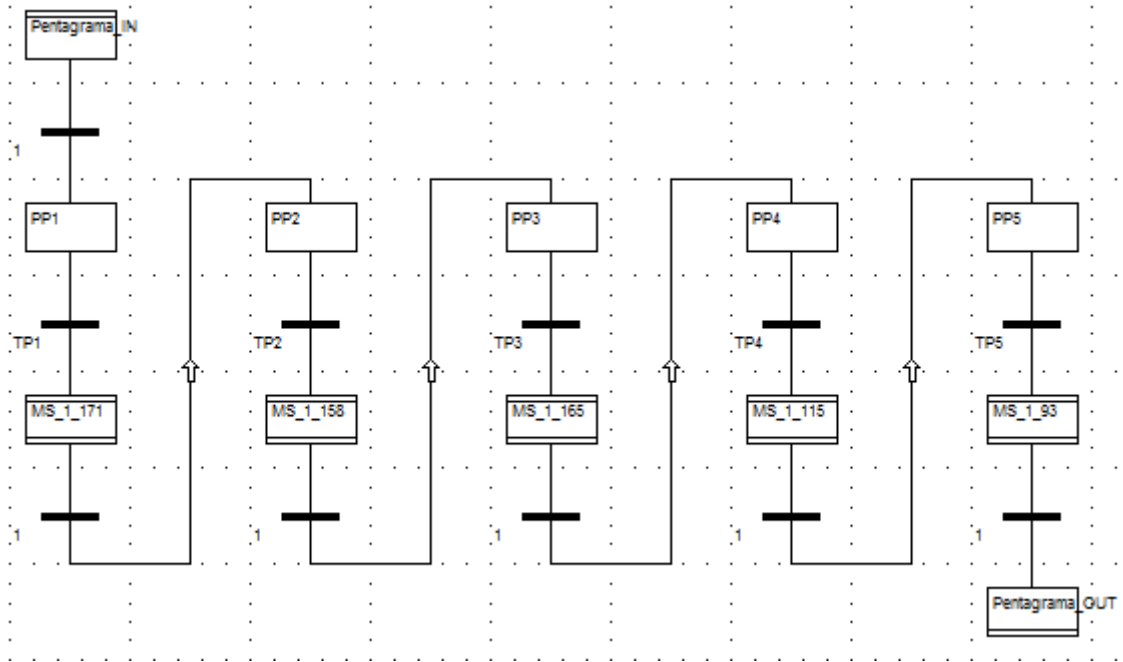


En este programa PC1, 2, 3 y 4 son las cuatro esquinas del cuadrado programadas por coordenadas en el esquema LD puntos. Como vemos en el primer bloque, insertamos el bit pulsando el botón Cuadrado, inmediatamente la máquina buscará el primer punto, después de esto el temporizador activará T3 para realizar unas acciones determinadas que explicaremos en breve. Y así sucesivamente se repetirá el proceso en PC2, PC3 y PC4.

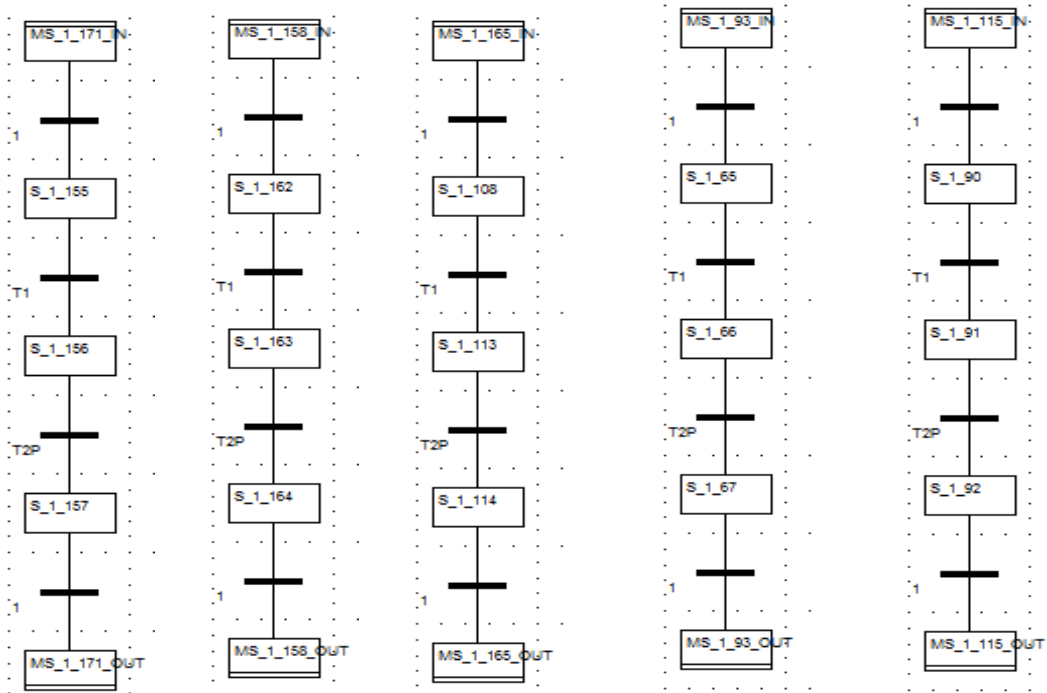


Aquí se muestran las acciones que realiza la máquina durante el programa de Cuadrado en cada punto que tiene marcado, con el primer bloque hemos entrado en el subprograma del cuadrado, después con el segundo bloque activamos las variables EV y TALADRO para activar el taladro y a la vez el descenso de éste mismo, después de activarse T1 que está gobernado por un temporizador y seguidamente se activa el ascenso del taladro. Una vez realizado esto procedemos al siguiente paso mediante T2 también gobernado por un temporizador que desactivará el taladro activando el siguiente bloque (esto se ha hecho para evitar atascos de la broca en la superficie a taladrar causado por la disminución de la velocidad de giro), finalmente, con el bloque out saldremos del subprograma para volver al programa Cuadrado y continuar con los demás puntos a perforar.

8.4.5.3. Pentagrama

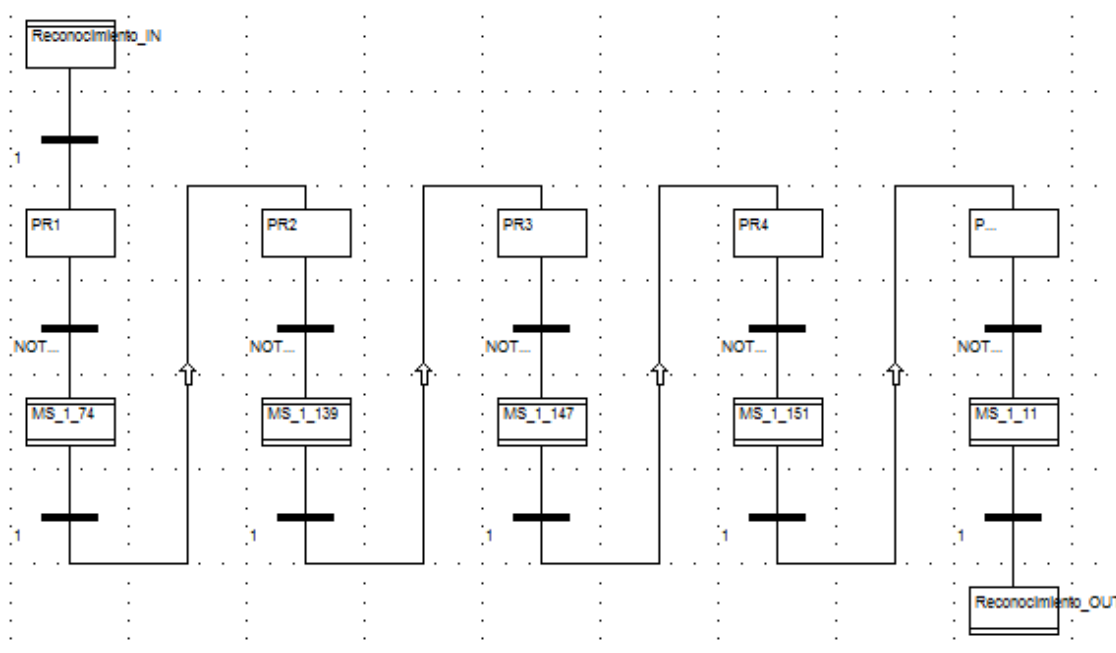


En este programa PP1, 2, 3, 4 y 5 son las cinco esquinas del pentágono programadas por coordenadas en el esquema LD puntos. Como vemos en el primer bloque, insertamos el bit pulsando el botón Pentágono, inmediatamente la máquina buscará el primer punto, después de esto el temporizador activará TP1 para realizar unas acciones determinadas que explicaremos en breve. Y así sucesivamente se repetirá el proceso en PP2, PP3, PP4 y PP5.

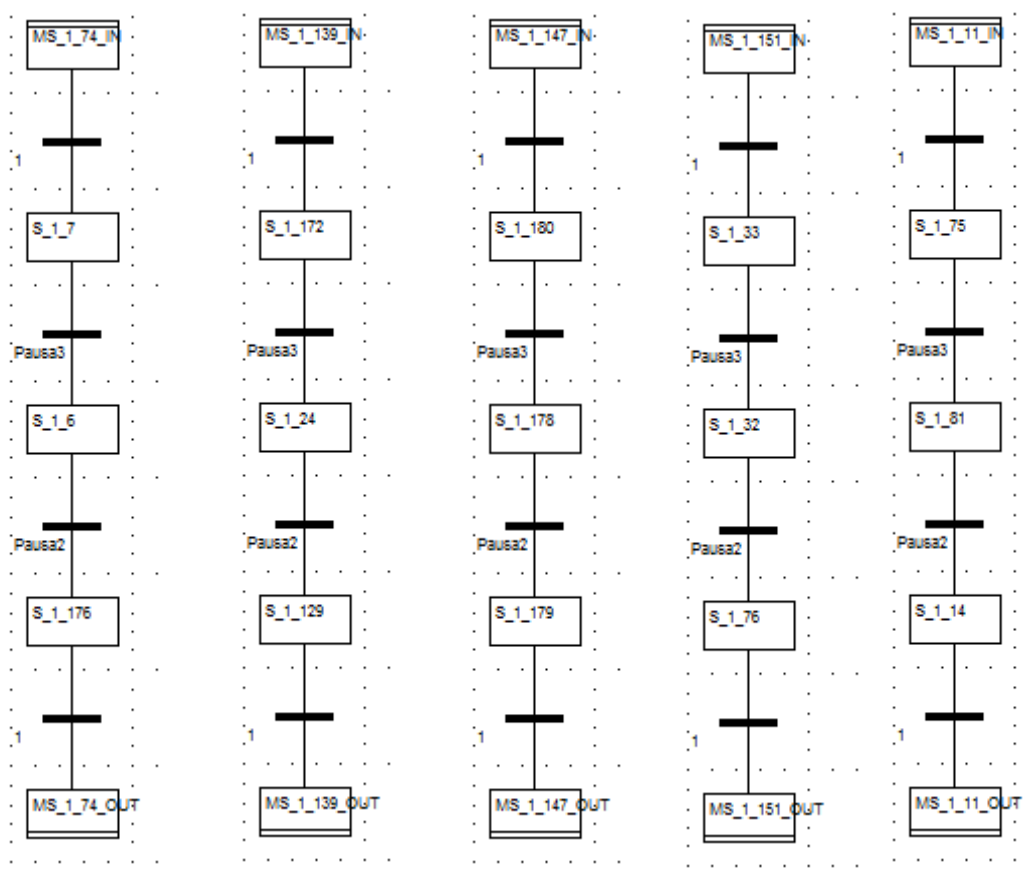


Aquí se muestran las acciones que realiza la máquina durante el programa de Pentágono en cada punto que tiene marcado, con el primer bloque hemos entrado en el subprograma del cuadrado, después con el segundo bloque activamos las variables EV y TALADRO para activar el taladro y a la vez el descenso de éste mismo, después de activarse T1 que está gobernado por un temporizador y seguidamente se activa el ascenso del taladro. Una vez realizado esto procedemos al siguiente paso mediante T2P también gobernado por un temporizador que desactivará el taladro activando el siguiente bloque (esto se ha hecho para evitar atascos de la broca en la superficie a taladrar causado por la disminución de la velocidad de giro), finalmente, con el bloque out saldremos del subprograma para volver al programa Pentágono y continuar con los demás puntos a perforar.

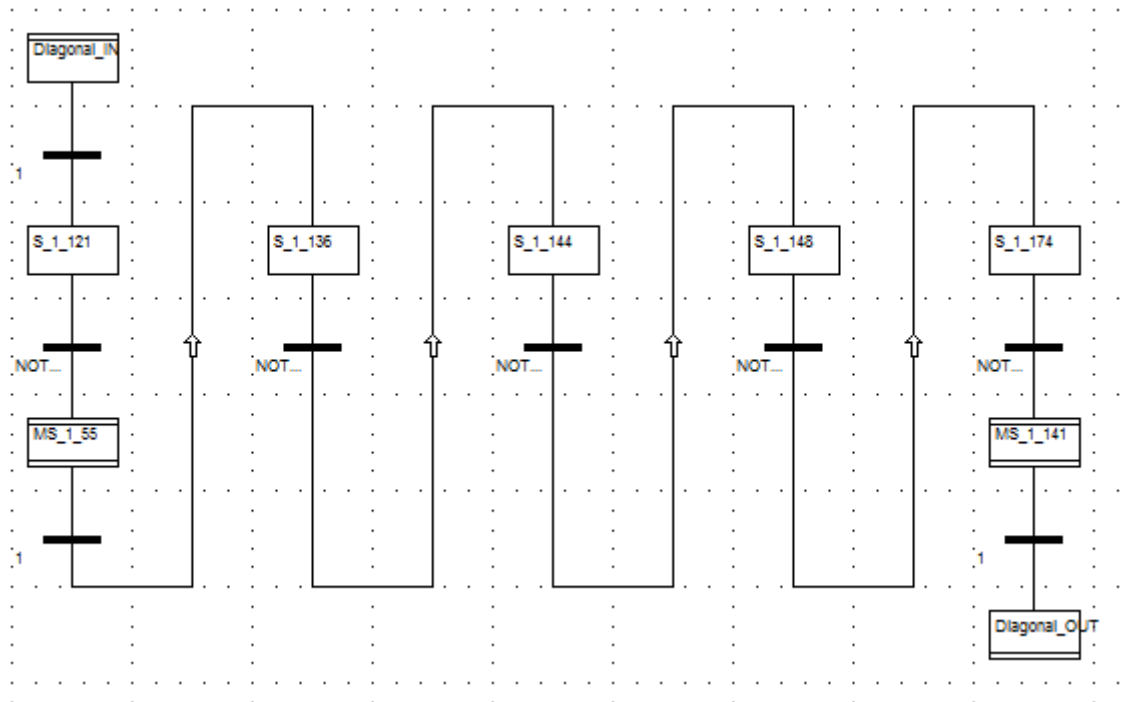
8.4.5.4. Reconocimiento



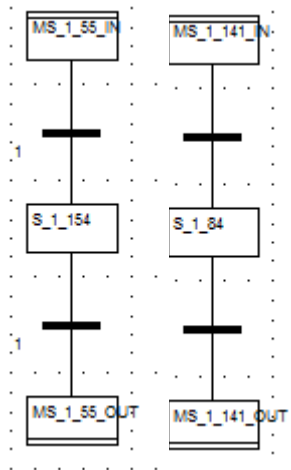
Este programa se ha creado como programa de prueba y mantenimiento y poder comprobar el correcto funcionamiento y calibrado de la máquina. Esto lo realiza marcando las cuatro esquinas de la zona máxima de trabajo donde puede taladrar la máquina. Es muy parecido al programa del pentágono a excepción de que el taladro no se activa, pero si realiza el descenso.



8.4.5.5. Diagonal



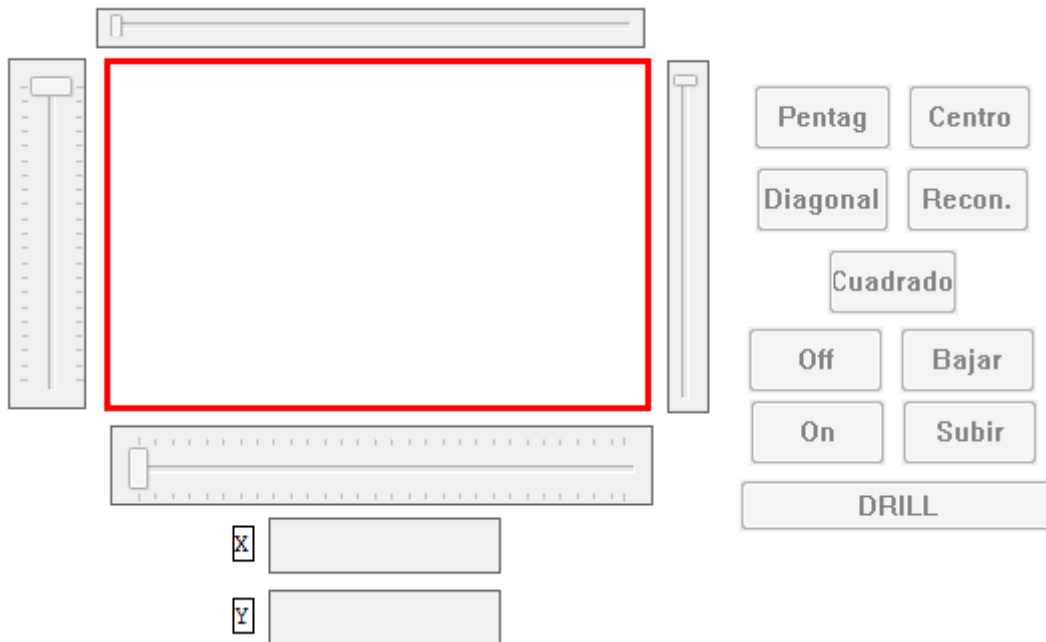
Con este programa, hemos marcado dos coordenadas que son los puntos de dos esquinas opuestas de la zona de perforación, también es una función de mantenimiento para comprobar que la máquina está bien coordinada.



8.5. Pantalla Unity PRO XL v8.0

Esta pantalla se ha creado para manejar el autómata a través de unos botones, cada uno con una variable determinada para posibilitar su activación o desconexión.

También se ha colocado dos barras deslizadoras y unos visualizadores numéricos.



Funciones de los elementos en pantalla

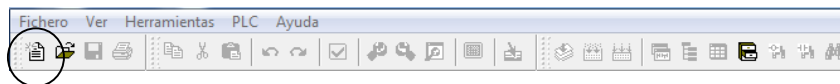
1. La barra deslizador horizontal está enlazada a la variable Px que es la encargada de la posición del eje x. Se moverá bajo nuestra acción o también seguirá el movimiento de los puntos perforados con el los programas SFC secundarios.
2. La barra deslizador vertical está enlazada a la variable Py que es la encargada de la posición del eje y. Se moverá bajo nuestra acción o también seguirá el movimiento de los puntos perforados con el los programas SFC secundarios.
3. El visualizador x también va enlazado a la variable Px, nos permitirá la visualización de la posición de la máquina en el eje x, además nos posibilita el manejo de la máquina mediante la introducción de las coordenadas.
4. Botón On activa la variable IR.
5. Botón Off desactiva la variable IR.
6. Botón Bajar activa la variable EV.

7. Botón Subir desactiva la variable EV.
8. Botón DRILL activa/desactiva la variable TALADRO.
9. Botón Pentag activa una vez la variable PENTAGONO
10. Botón Centro activa una vez la variable Centro.
11. Botón Cuadrado activa una vez la variable CUADRADO.
12. Botón Diagonal activa una vez la variable DIAGONAL.
13. Botón Recon. activa una vez la variable RECONOCIMIENTO.

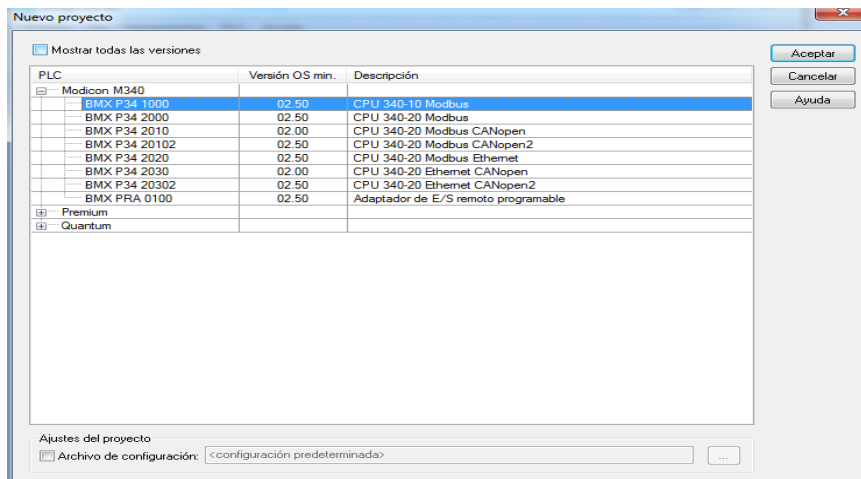
8.6. Creación de un programa con Unity PRO

En esta sección explicaremos como se ha hecho el proyecto paso a paso:

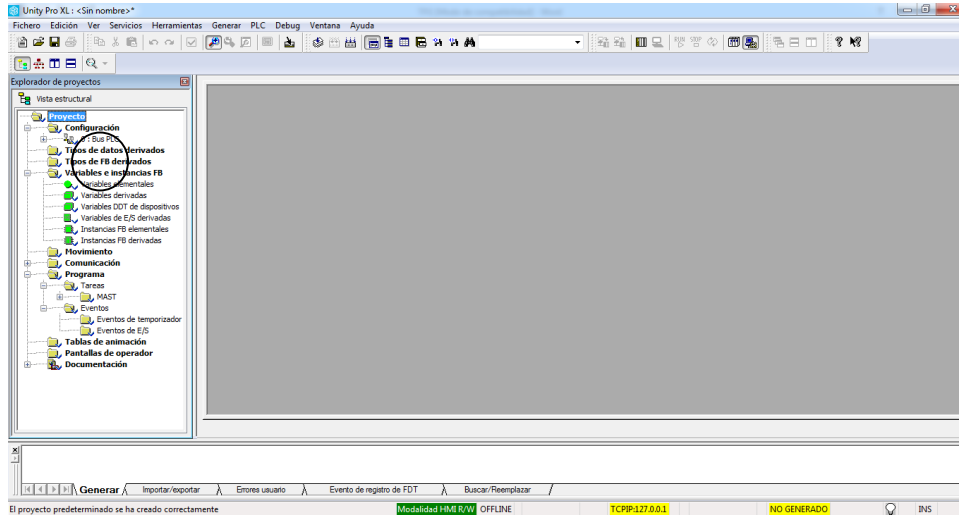
1. Abrir el Unity PRO XL 8
2. Pulsa el icono señalado



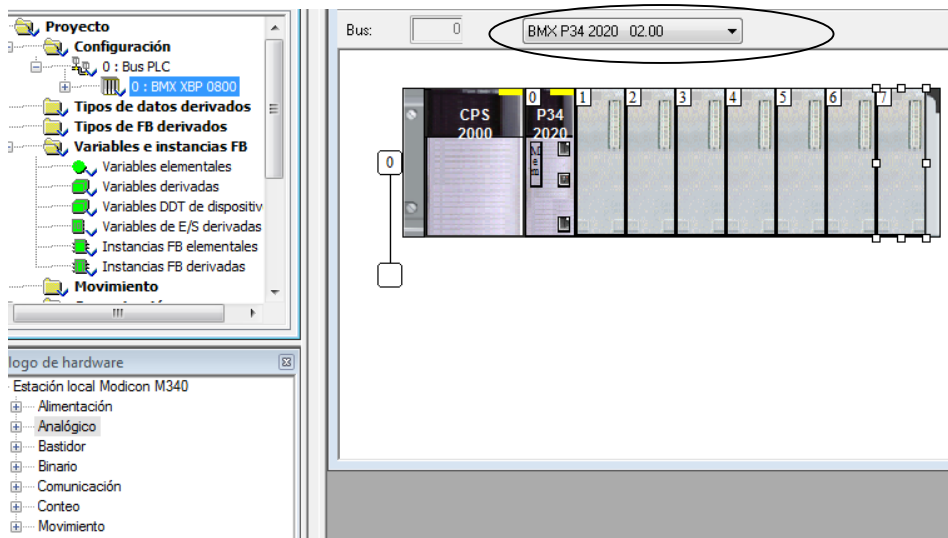
3. Debemos escoger el tipo de PLC, en nuestro caso el Modicon M340 señalado en la figura y aceptamos.



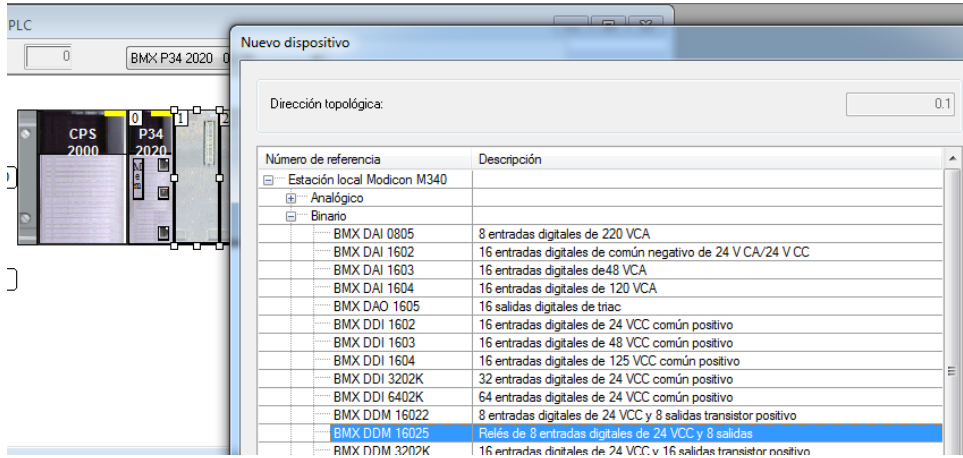
4. Aparecerá esta ventana y deberemos configurar todos los dispositivos que vamos a necesitar controlar



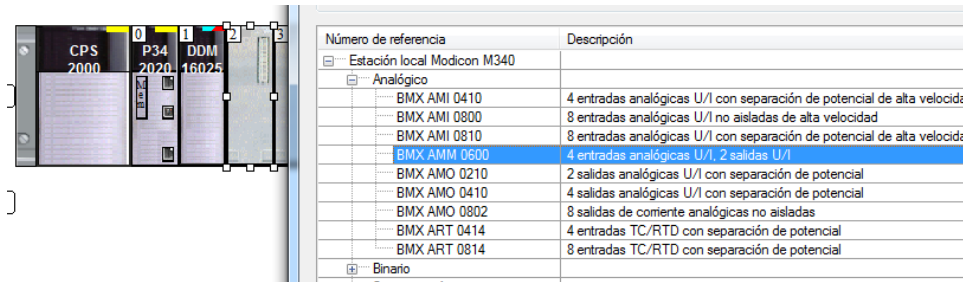
5. Escogemos el BMX P34 2020 2.00 debido a que tiene control mediante PLC y ETHERNET, muy importante ya que permite un control del PLC vía PC o través de la red.



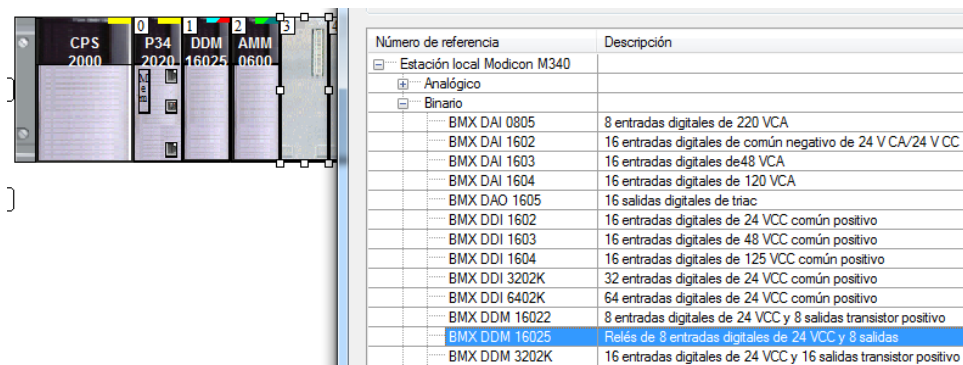
6. Haciendo doble click sobre el módulo 1 se abrirá un menú



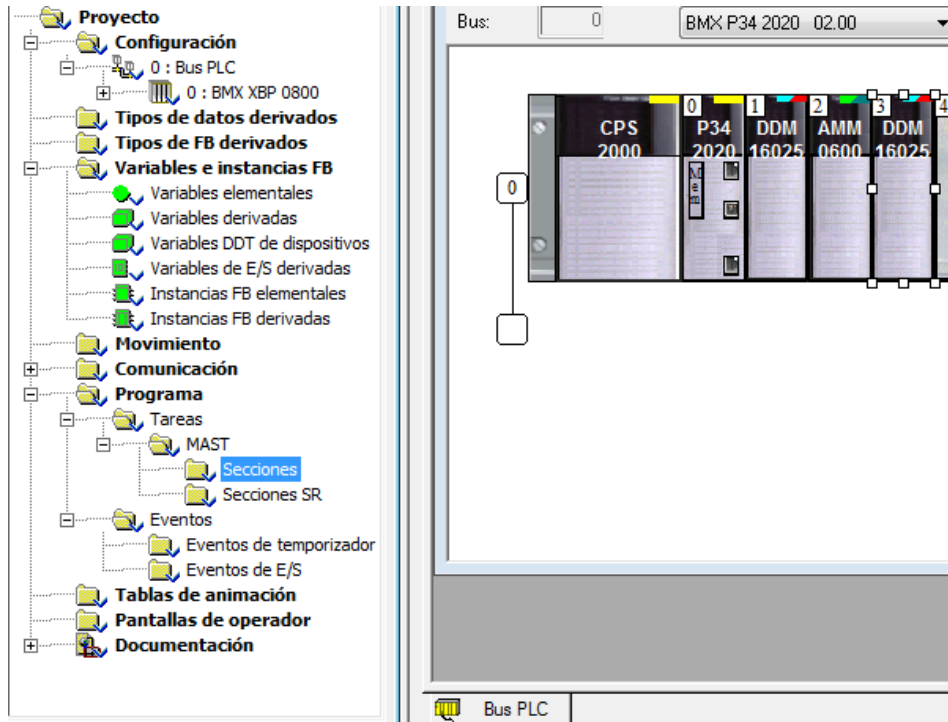
7. Añadimos el módulo 2



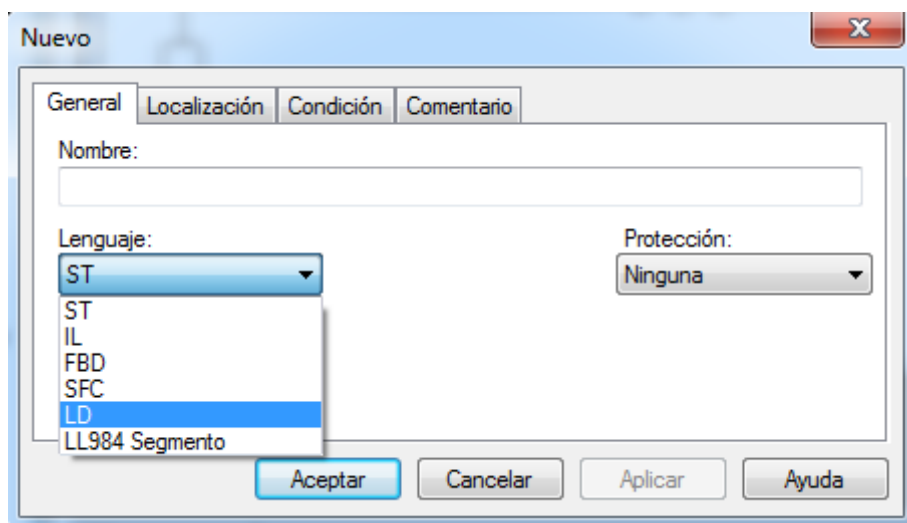
8. Añadimos el módulo 3



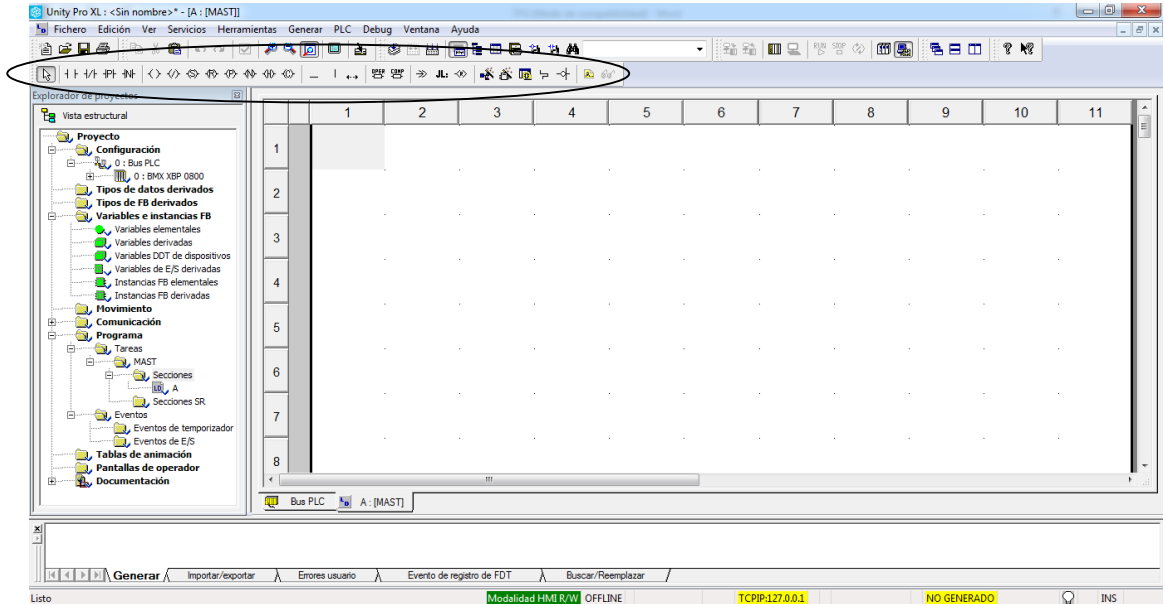
9. Con todo configurado, escogemos crear una nueva sección, con el botón derecho del ratón escogemos “nueva sección...”



10. Aparece esta ventana donde podemos escoger el tipo de lenguaje de programación, primero el LD

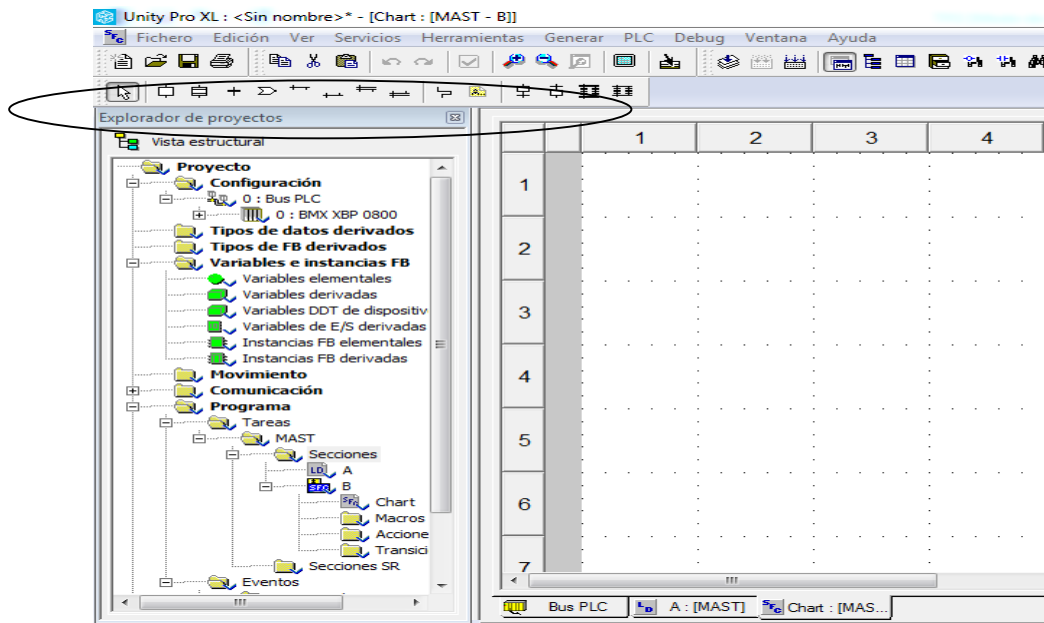


11. Nos llevará al editor LD con los elementos que se han explicado en apartados anteriores

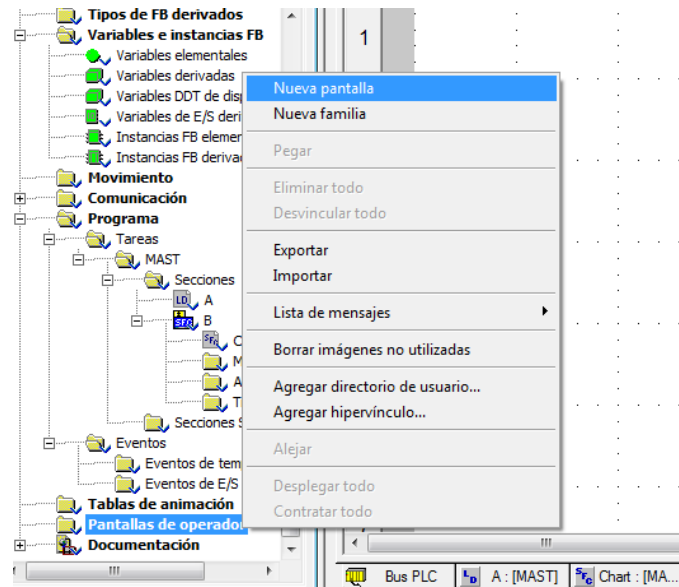


A partir de aquí ya empezamos a diseñar el programa LD

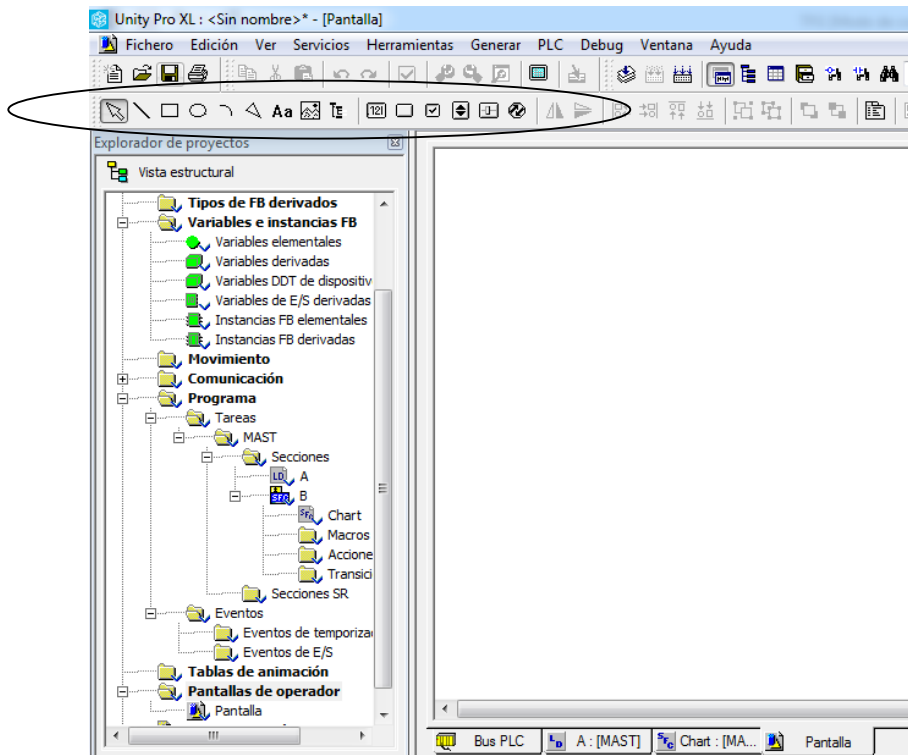
12. Si escogemos el lenguaje SFC, nos aparecerá el siguiente editor SFC con los diferentes elementos utilizados para diseñar el programa SFC





13. Para crear la pantalla solo hay que escoger nueva pantalla con el botón derecho del ratón en el lugar señalado en la figura



14. Aparecerá el editor de pantalla



15. Finalmente, cuando esté todo diseñado, se guarda y se transfiere, esto se hace primero Conectando el PLC al PC con , y seguidamente transferimos con .

Para activar el programa del PLC transferido deberemos pulsar RUN.

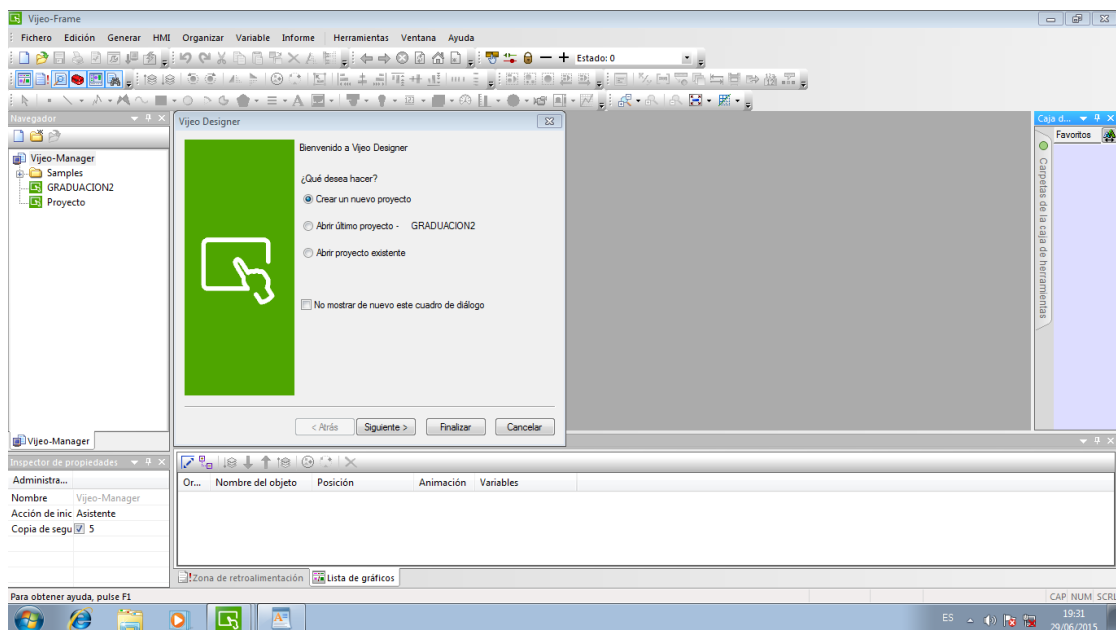
9. Programa Vijeo Designer v6.1

9.1. Elementos de Vijeo Designer

Este programa se ha escogido debido a que es mucho más manejable a la hora de representar gráficamente, permite mayor flexibilidad que el programa Unity PRO y además es el que ha creado Schneider para trabajar con los paneles avanzados Magelis HMI STU 855.

9.2. Realización del proyecto

Nada más abrir el Vijeo nos aparecerá la siguiente ventana:



Aquí podemos escoger si abrir un nuevo proyecto o uno ya existente (los ya existentes se encuentran en la carpeta Vijeo-manager, es la única manera para insertar un proyecto Vijeo creado en otra terminal).

Crear un nuevo proyecto

Nombre del proyecto a crear

Nombre del proyecto Proyecto

Descripción o comentario

Tipo

Proyecto con un sólo destino

Proyecto con 2 destinos

Contraseña del proyecto

Ingrese contraseña

Confirme contraseña

Recordar (opcional)

< Atrás Siguiete > Finalizar Cancelar

En esta ventana se nos pide que insertemos el nombre escogido del proyecto, si queremos insertar alguna contraseña para abrir el proyecto, y finalmente la cantidad de destinos al que va destinado el proyecto, porque si queremos podemos trabajar con varios autómatas a la vez. En nuestro caso se ha escogido un solo destino, y será nuestro panel Magelis.

Como disponemos de un panel STU 855, lo tendremos que indicar en el programa para posibilitar la comunicación, también podemos seleccionar un nombre de destino, como se puede ver en la figura siguiente.

Crear un nuevo proyecto

Nombre del proyecto a crear

Nombre del proyecto

Destino : 1/1

Nuevo proyecto/destino

Nombre del destino

Tipo de destino

Modelo

< Atrás **Siguiente >** Finalizar Cancelar

El próximo paso es el siguiente, aunque no haremos nada de momento, ya se configurarán todos los pasos más tarde.

Crear un nuevo proyecto

Nombre del proyecto a crear

Nombre del proyecto

Destino : 1/1

Configuración de destino

Asignar la siguiente dirección IP

Dirección IP

Máscara subred

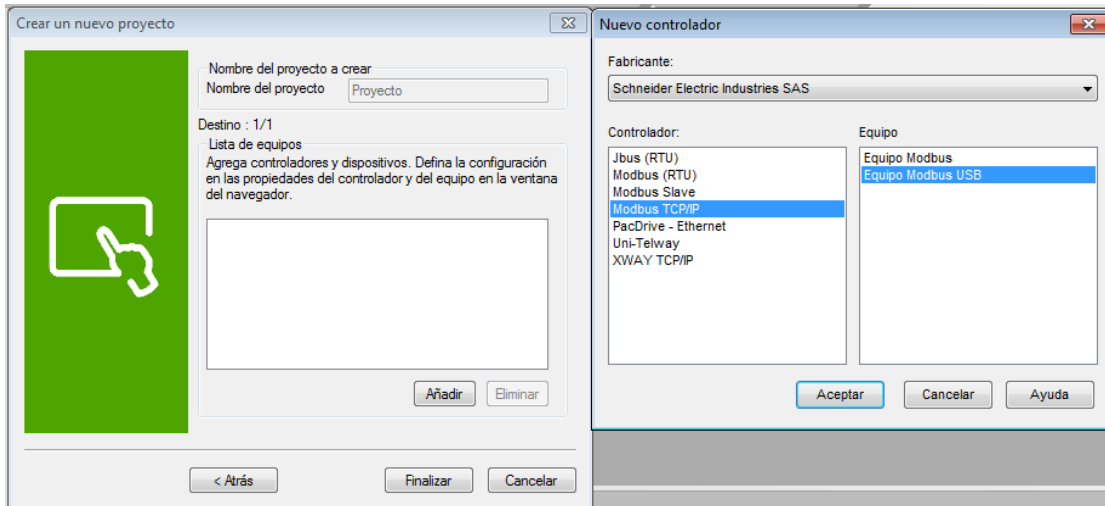
Puerta enlace predeterminada

Habilitar auditoría

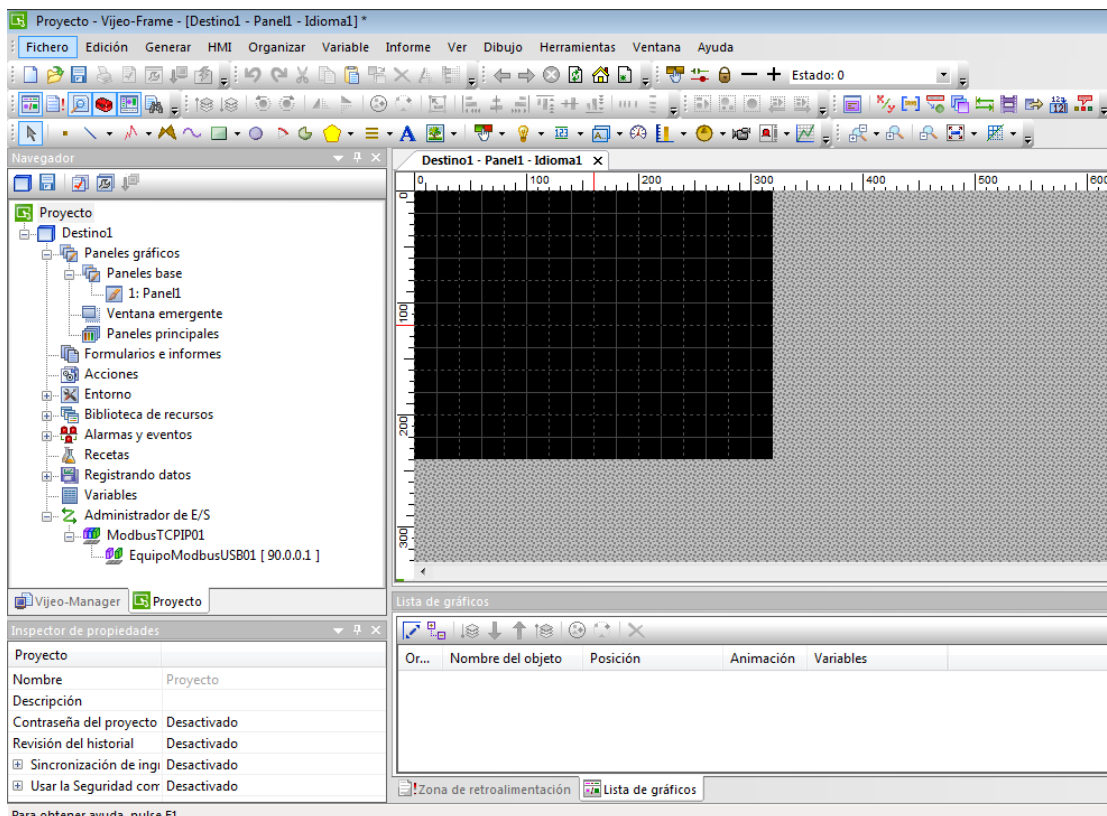
Se requiere una configuración adicional para usar esta función. Haga clic en la ayuda y revise la configuraciones necesarias.

< Atrás **Siguiente >** Finalizar Cancelar

En la siguiente ventana debemos escoger el tipo de controlador, el cual controlará nuestra maqueta. En nuestro caso seleccionamos El indicado en la imagen, ya que utilizaremos una conexión USB entre el panel y el Modicom.



Ahora ya hemos configurado nuestro proyecto y podemos pasar a diseñar la parte gráfica.

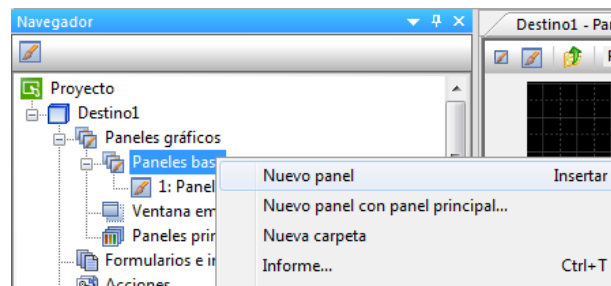


Los pasos a seguir ahora son:

- a. Realizar los paneles gráficos.
- b. Guardar el proyecto.
- c. Ejecutar una comprobación de errores.
- d. Compilar el proyecto.
- e. Instalar V.D. Runtime en la máquina de destino.
- f. Descargar el proyecto en la máquina de destino.
- g. Ejecutar el proyecto.

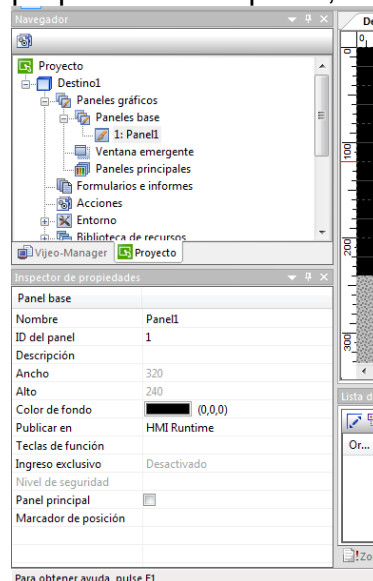
9.2.1. Gestión de los paneles

Un panel es una ventana en la que se pueden colocar objetos (interruptores, lámparas, dibujar elementos). El panel creado se convierte en la pantalla que aparece en las máquinas de destino. Para crear un nuevo panel, iremos a la ventana del navegador, haremos clic con el botón derecho en los paneles base y seleccionaremos 'nuevo panel' (también podremos copiar o eliminar paneles del mismo modo).

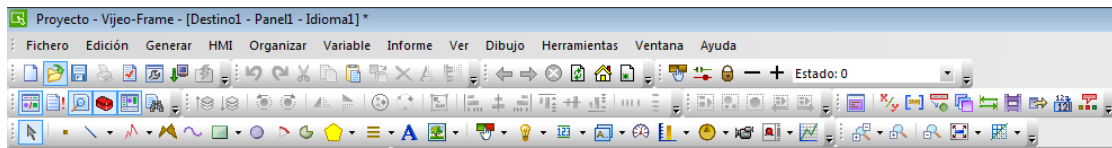


Propiedades de los paneles

Cuando se selecciona un panel en el navegador. En el inspector aparecen las propiedades del panel, donde se pueden configurar.



Las opciones gráficas de las que disponemos son las siguientes:



Como se puede apreciar, hay muchísimas opciones para escoger, así que en este caso solo explicaremos las utilizadas para la realización del trabajo de final de grado.

- Herramientas de dibujo clásicas



- Objetos gráficos



- Herramientas de opciones de selección



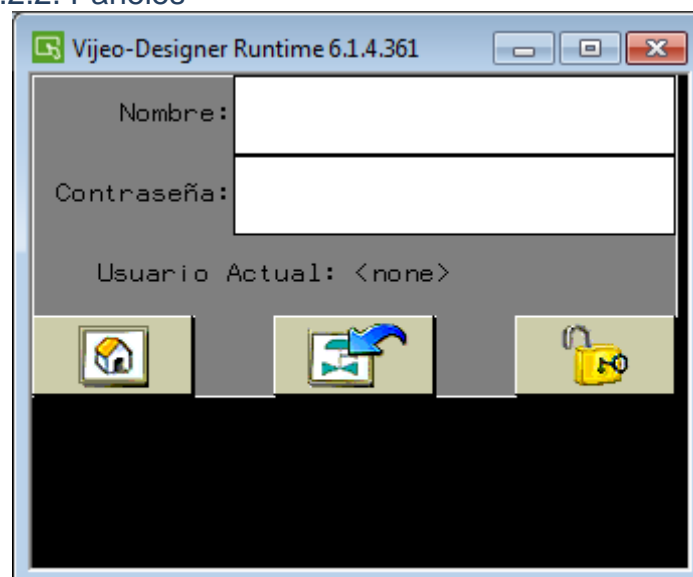
- Herramienta de edición

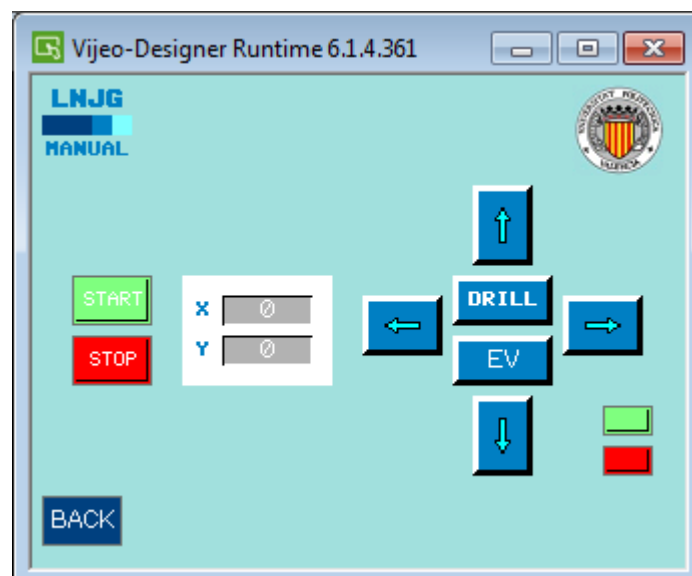
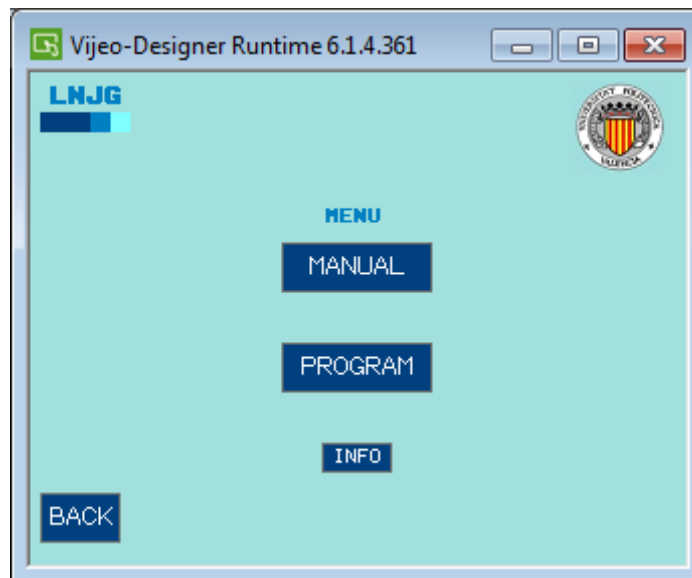
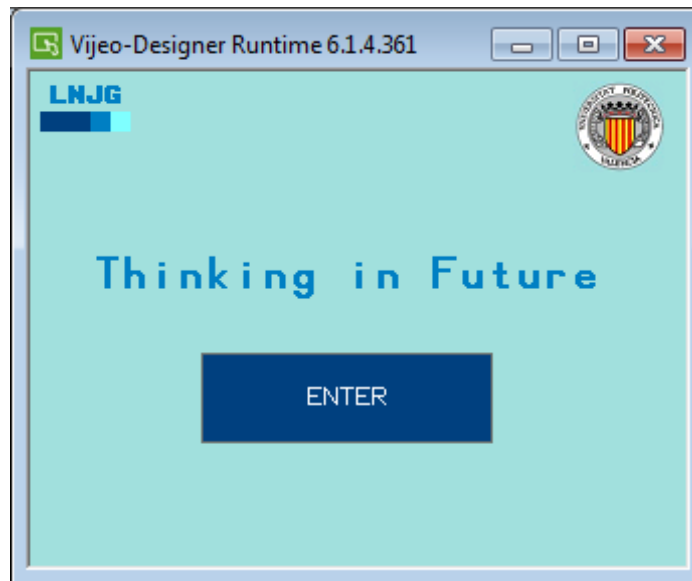


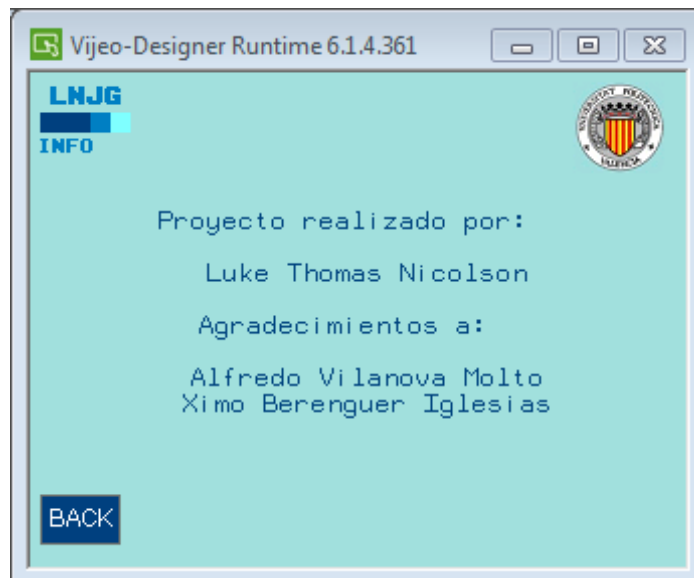
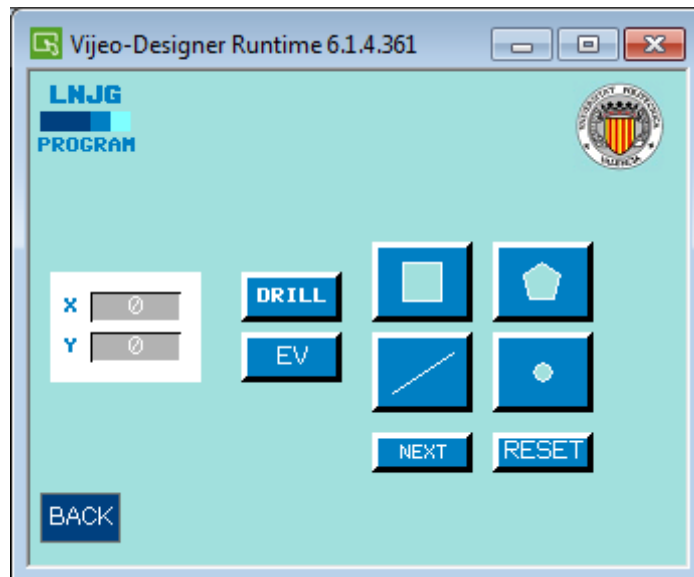
- Herramienta de organización



9.2.2. Paneles

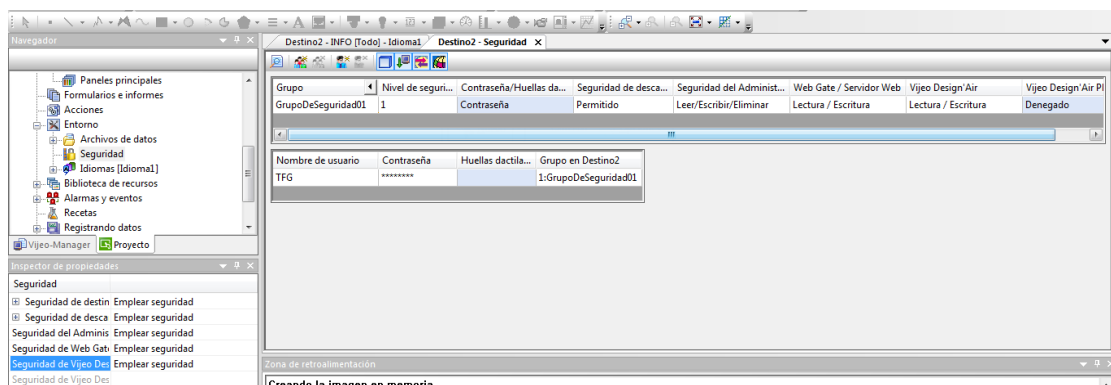




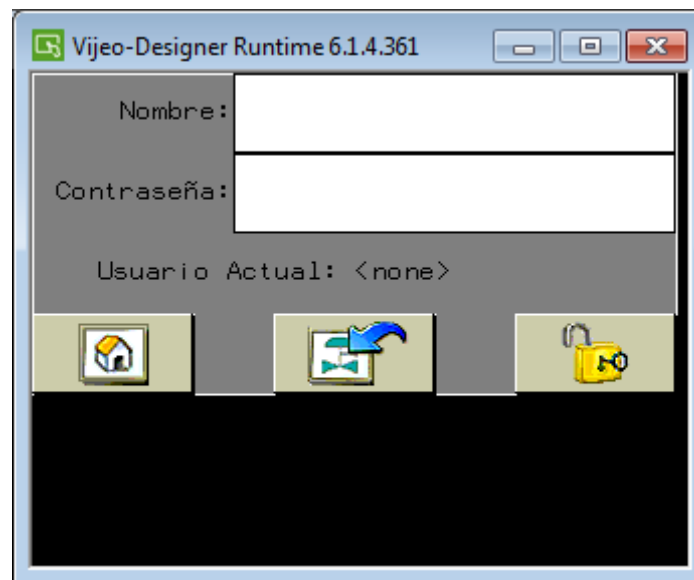


9.3. Creación del programa

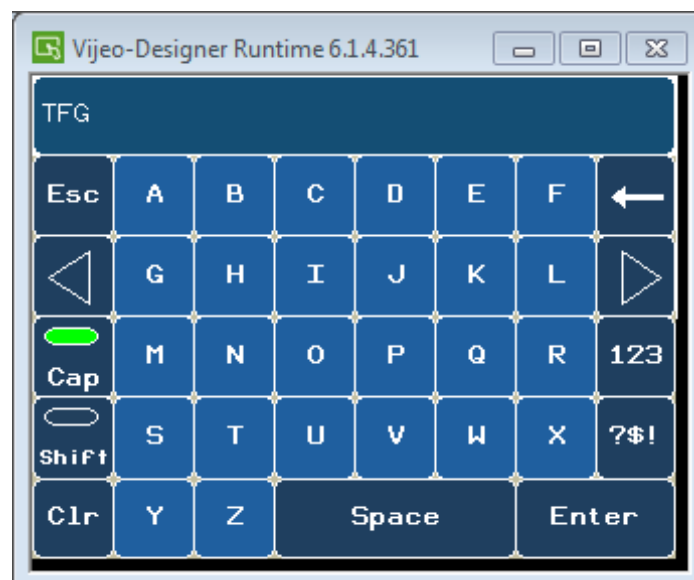
9.3.1. Añadir seguridad al control de la máquina



En el navegador hay que seleccionar Entorno/ Seguridad para que nos aparezca en el inspector la posibilidad de habilitar o deshabilitar la seguridad y en la ventana principal insertamos los usuarios, contraseñas y aplicaciones que queremos proteger. Al aplicar seguridad, cuando se encienda el HMI Magelis nos aparecerá lo siguiente:



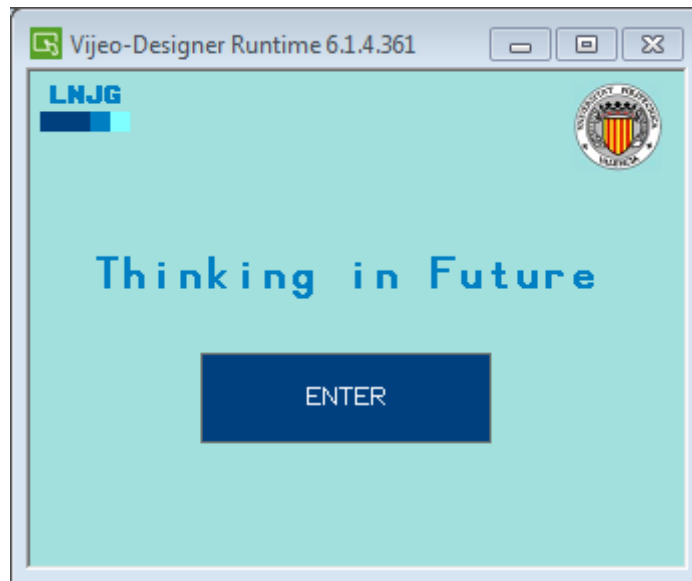
Al seleccionar cualquier apartado aparecerá un teclado numérico.



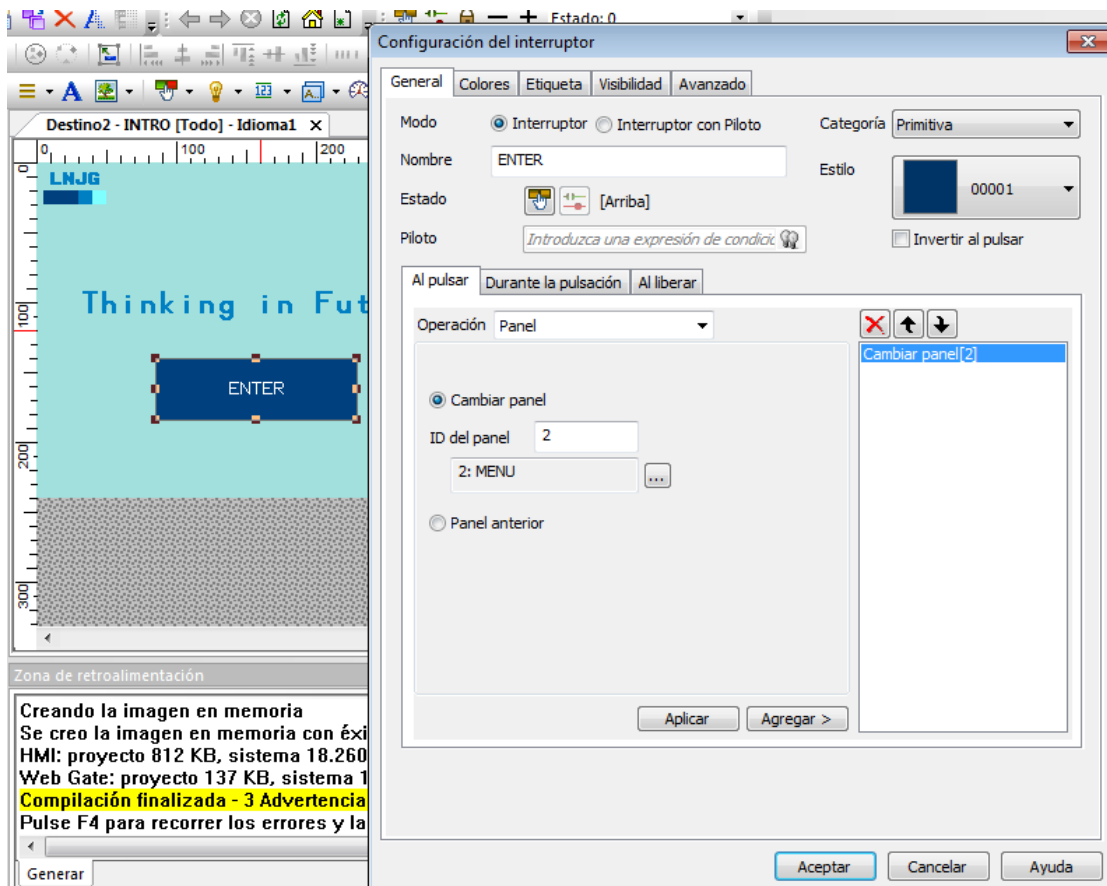
Aquí insertaremos:

Nombre: TFG
Usuario: TFG

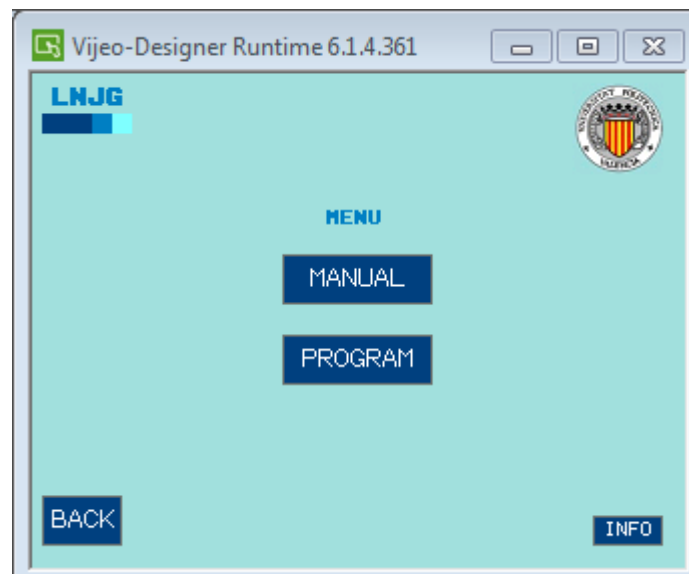
Y seleccionamos la tecla Enter. Una vez hecho esto pasaremos al panel principal.



En este panel tenemos un interruptor, uno de los objetos gráficos del programa. A este interruptor se le ha programado la acción de cambiar al panel 2 de menú como se ve en la ventana de configuración del interruptor.



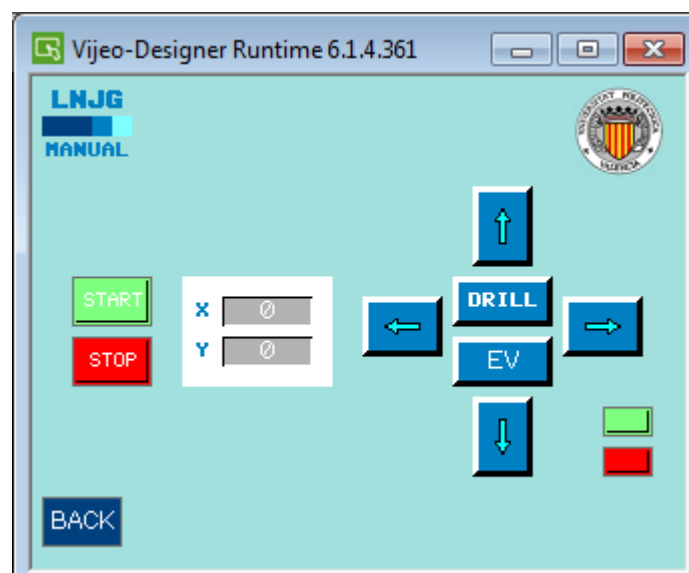
En el panel 2 aparece lo siguiente:



Se muestra el botón MANUAL para ir al panel 3 y así utilizar el taladro de forma manual, y también el botón PROGRAM para manejar el taladro con los programas grabados.

El botón BACK permite volver al panel principal y el botón info se ha creado para ir al panel 5 para ver la información del SCADA diseñado.

Aquí se muestra el panel 3 MANUAL



En este panel disponemos del botón START y STOP, que activan y desactivan nuestra variable IR, la encargada de activar el movimiento del taladro.

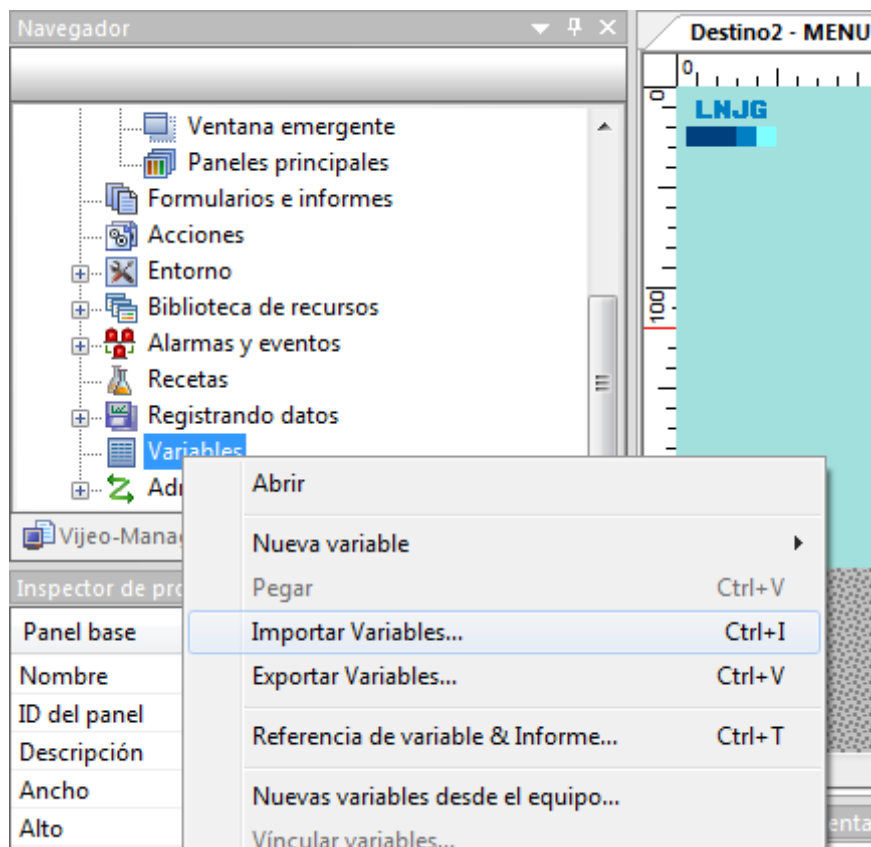
El visualizador, nos permite además de observar la posición del taladro en el eje x e y, insertar coordenadas con el teclado numérico para que se desplace a dicha posición.

A la derecha se observa la cruz de movimiento que se activa apretando el botón verde de la derecha, y además están el botón DRILL que activa el taladro y EV el cual activará el pistón neumático para comenzar el descenso del taladro.

Como se ha indicado antes, se ha enlazado la variable IR del programa Unity PRO con un interruptor, además de otras variables necesarias para controlar todo el proceso.

Esto se ha hecho de la siguiente manera:

Primero debemos de guardar en un formato .CSV el proyecto en el Unity PRO, una vez hecho esto, buscamos el apartado de variables en nuestro navegador. Pulsamos el botón derecho y escogemos importar variables.



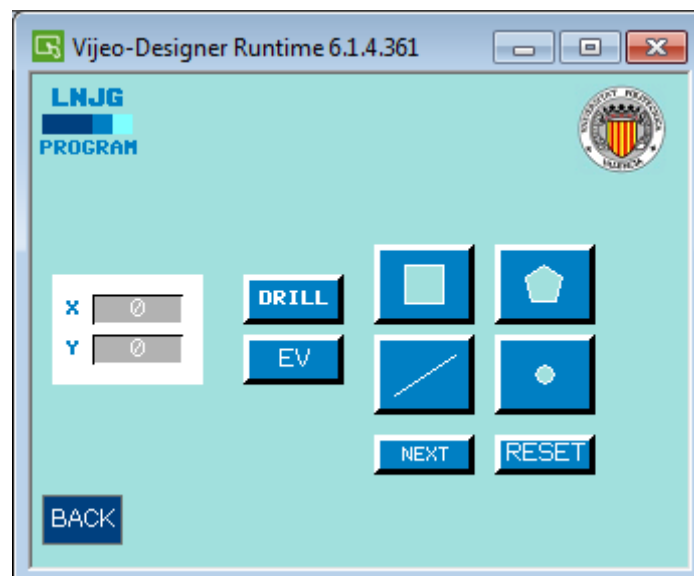
Se nos abrirá un menu en el cual debemos seleccionar nuestro proyecto de Unity PRO guardado recientemente, sigientemente aparecerá una ventana que nos permite escoger las variables que queremos, en nuestro caso escogeremos seleccionar todo y después vincular.

Lo siguiente que nos muestra es una tabla con todas las variables vinculadas de nuestro proyecto Unity. Aquí deberemos de modificar las variables necesarias para que sean de lectura/escritura para poder manejar todas las variables desde el Magelis, nuestro dispositivo android y también desde el servidor de webgate.

Variable	Nombre	Descripción	Dimensión de matriz	Tipo de datos	Origen	Compartiendo	Grupo de escaneo	Dirección de dispositivo	Dirección indirecta
	EcParabola3			BOOL	Externo	EquipoModbus...	%UV46.1472.8.255	Desactivado	Ninguno
	EcParabola4			BOOL	Externo	EquipoModbus...	%UV46.1473.8.255	Desactivado	Ninguno
	EcParabola5			BOOL	Externo	EquipoModbus...	%UV46.1474.8.255	Desactivado	Ninguno
	Ecuac			BOOL	Externo	EquipoModbus...	%UV46.1455.8.255	Desactivado	Ninguno
	Ecuacion			SFCSTEP_STATE	Externo	EquipoModbus...			
	Ecuacion_IN			SFCSTEP_STATE	Externo	EquipoModbus...			
	Ecuacion_OUT			SFCSTEP_STATE	Externo	EquipoModbus...			
	ELIPSE			BOOL	Externo	EquipoModbus...	%UV46.1419.8.255	Desactivado	Ninguno
	Emergencia			BOOL	Externo	EquipoModbus...	%UV46.1468.8.255	Desactivado	Ninguno
	ENTRADA_TALADRO			BOOL	Externo	EquipoModbus...	%UV46.1497.8.255	Desactivado	Ninguno
	Estado			UINT	Externo	EquipoModbus...	%UV46.1394.16.25	Desactivado	Ninguno
	EV			BOOL	Externo	EquipoModbus...	%Q0.3.18	Desactivado	Ninguno

Con este vínculo ya podemos controlar cualquier variable insertada en el autómata.

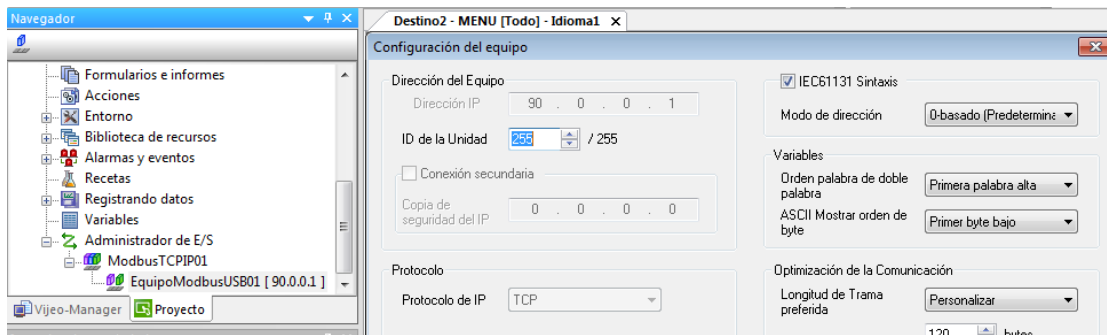
Pasamos al panel 4, el cual nos permite activar los programas explicados en el apartado del programa Unity PRO.



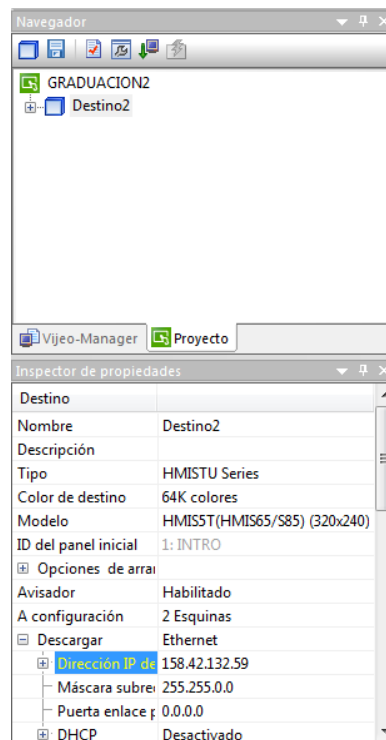
Volvemos a tener un visualizador, pero en este caso solo es para observar, en el modo programa no se pueden insertar coordenadas para evitar cualquier mal funcionamiento de la máquina. También vuelven a aparecer los botones DRILL y EV con las mismas funciones anteriormente explicadas. Aparecen un botón con un cuadrado y otro con un pentágono, estos serán los encargados de activar los programas de cuadrado y pentágono para así activar el movimiento de la máquina. El botón con la línea en diagonal activa el movimiento en diagonal que se controla el paso de cada punto con el botón NEXT. Finalmente el botón RESET se utilizaría para procesos de mantenimiento ya que activa el programa de reconocimiento.

El panel 5 se ha hecho como agradecimiento a las personas que me han ayudado a realizar este TFG, Alfredo Vilanova Molto y Ximo Berenguer Iglesias.

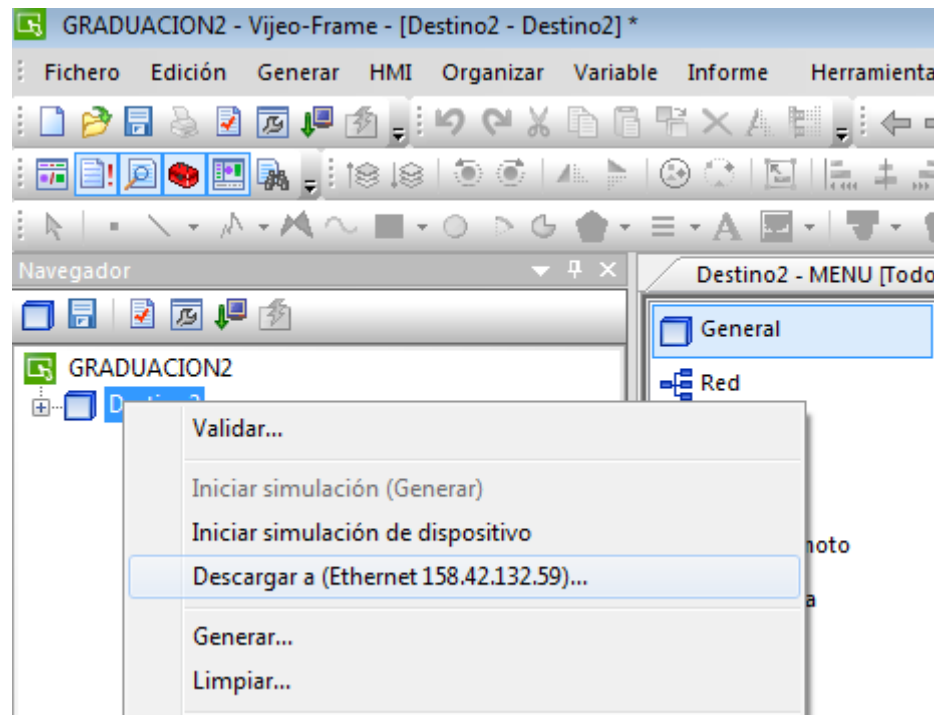
A continuación se explicará los pasos a realizar para conectar el autómeta. Hacemos doble clic sobre EquipoModbus y aparece la siguiente ventana. Debemos de seleccionar la casilla indicada.



Después de esto solo nos quedará insertar el programa Unity en el Modicom y ponerlo en modo RUN, finalmente, transferiremos el programa Vijeo al Magelis configurando su IP y seleccionando la descarga vía ethernet. Se muestra en la siguiente ventana.



Ya solo queda descargar el programa, se hace de la siguiente manera:



El momento que descargamos, Vijeo hace un análisis de errores en la programación, si hubiera alguno, Vijeo nos lo mostrará con el botón F4.

Ahora ya podremos manejar la máquina a través de la pantalla Magelis.

10. Vijeo Webgate

Vijeo Webgate es el servidor que nos permite manejar los paneles diseñados con Vijeo designer que controlan nuestro autómata. En el siguiente apartado se explicará como utilizarlo.

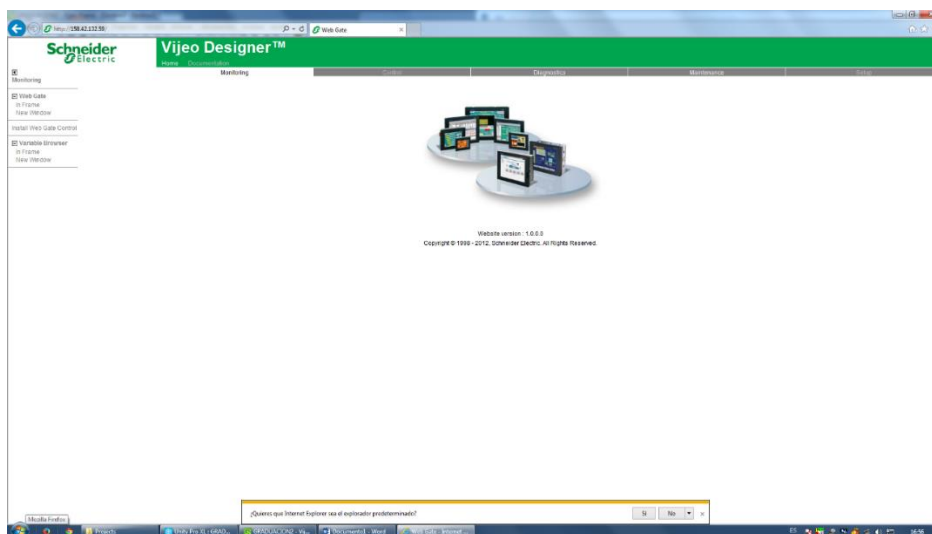
10.1. Creación del Webgate

Lo primero que hay que hacer es asegurarnos que la conexión Ethernet está disponible y además deberemos de saber la IP de nuestra pantalla Magelis.

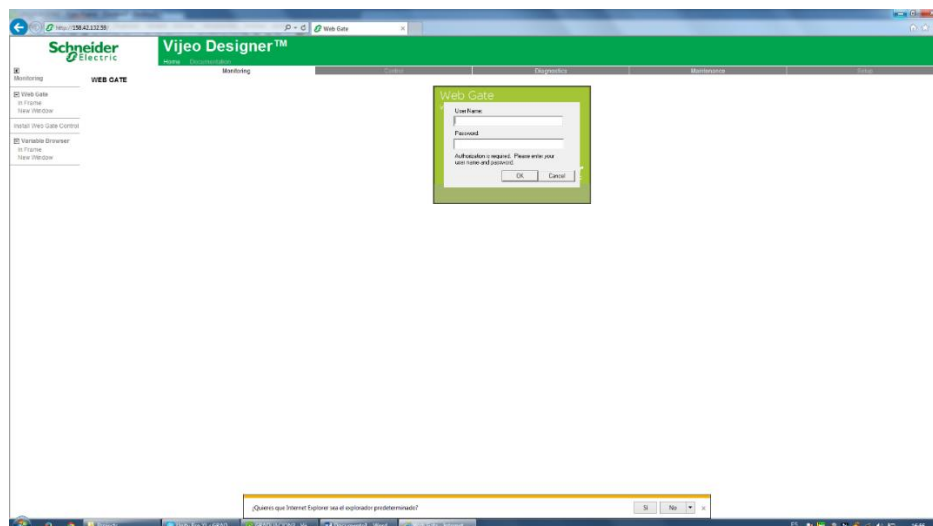
Después debemos asegurarnos que el runtime de vijeo webgate está instalado, sino no conseguiremos visualizar los paneles diseñados.

Empezaremos abriendo el navegador de Internet Explorer, ya que es el único que dispone de lo necesario para reproducir los paneles. Una vez

hecho esto, insertamos la IP en el navegador y nos aparecerá la siguiente ventana.

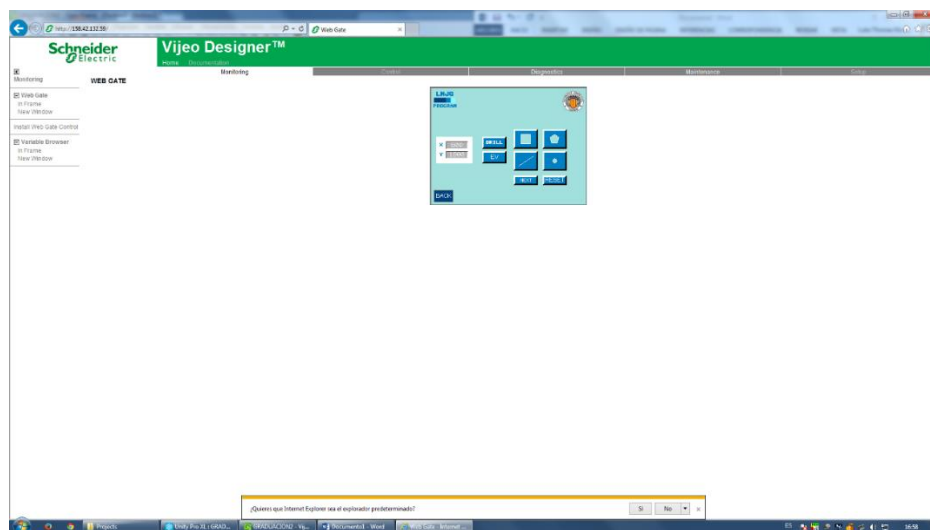
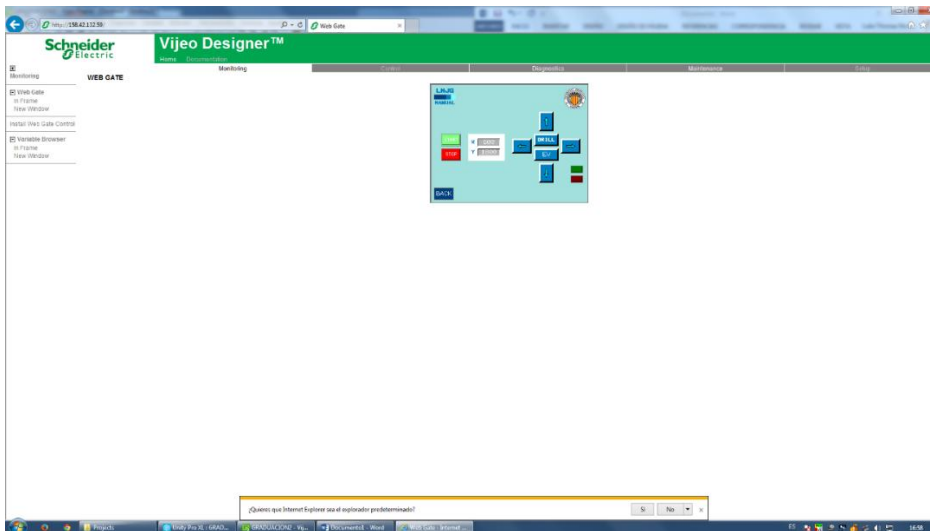
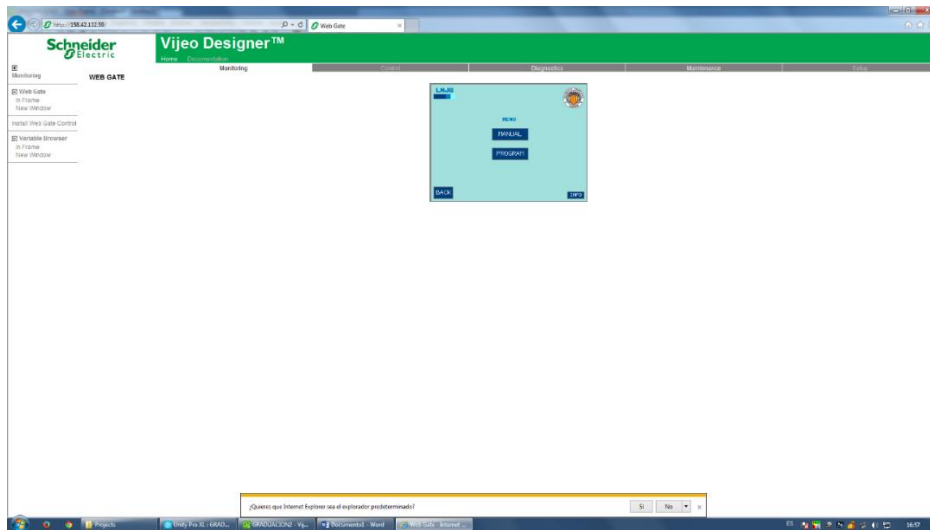


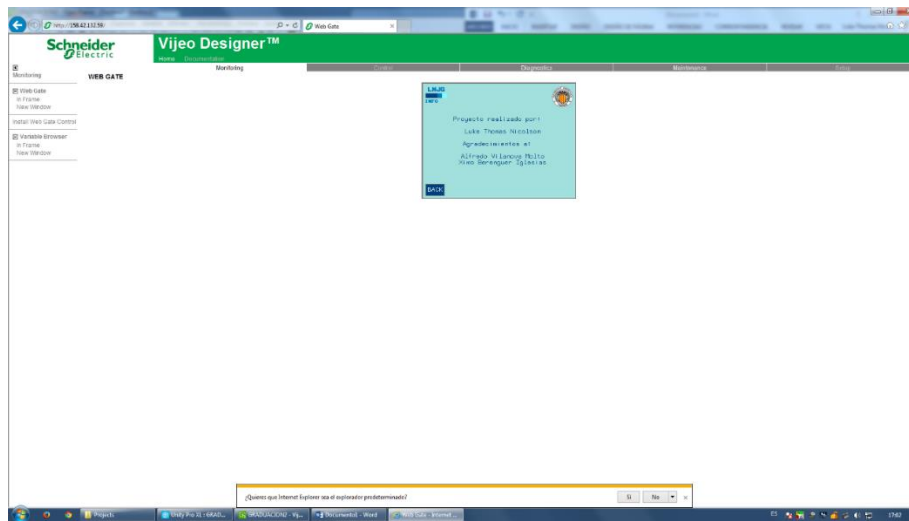
Iremos a monitoring y seguidamente entraremos en Webgate/in frame, seguidamente pasaremos a la siguiente ventana, en la cual nos aparecerá 2 huecos, uno para rellenar el usuario y otro para la contraseña (iguales que las asignadas en el Vijeo Designer para activar la seguridad).



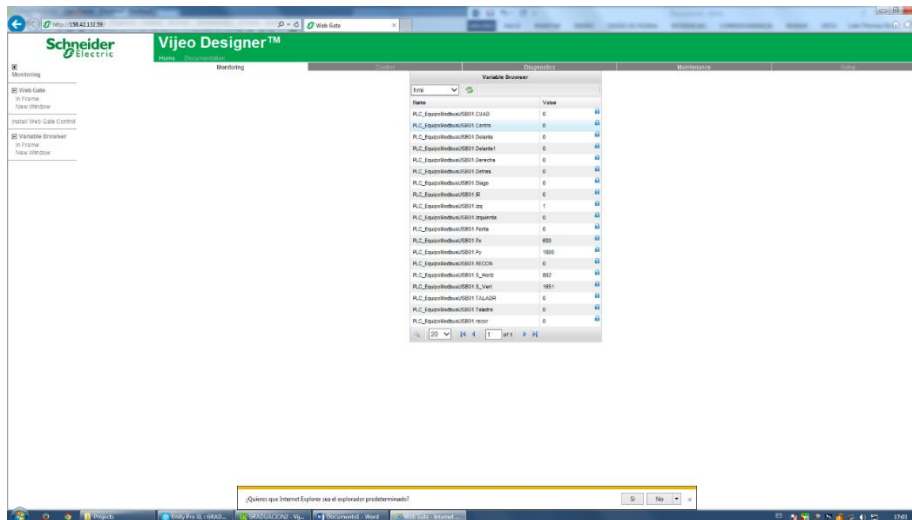
Alcanzado este paso, inmediatamente se nos mostrará nuestro panel de Vijeo en el navegador Internet Explorer como se muestra en la siguiente figura.

A partir de aquí ya podemos manejar el panel como se ha explicado en el apartado de Vijeo Designer.

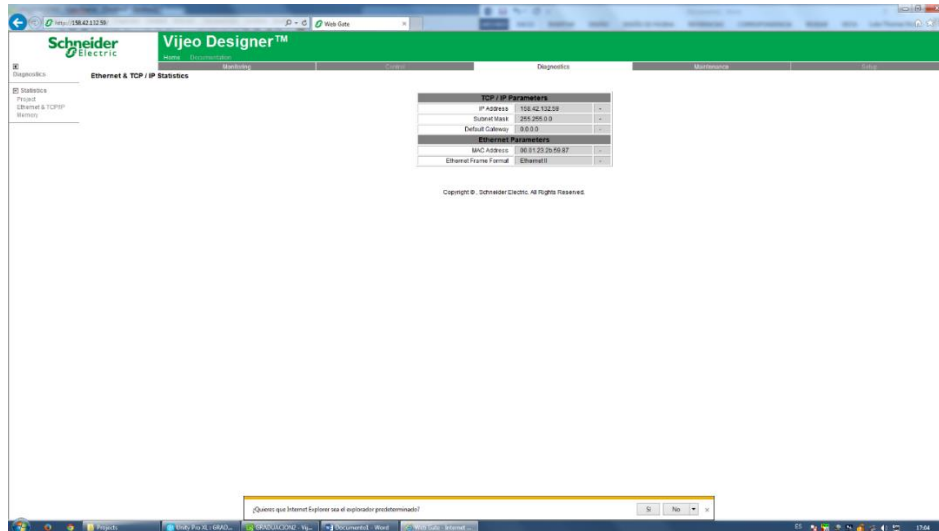




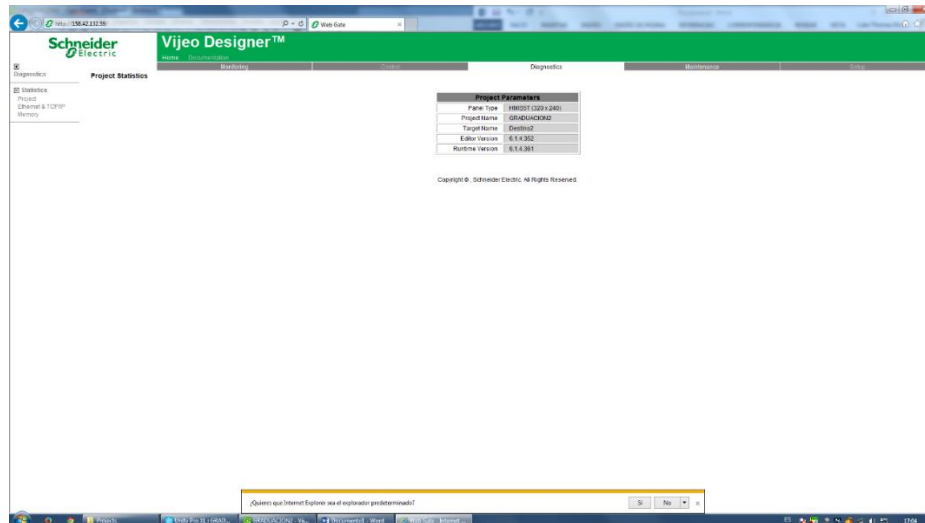
En el apartado Variable browser, si seleccionamos in Frame, iremos a una tabla en la que están todas las variables que se han asignado al panel para permitir el manejo correcto del taladro. Esta ventana es muy útil si no disponemos de internet explorer, ya que no nos permite ver la representación gráfica de nuestros paneles, pero si nos permite cambiar sus valores, y así controlar de igual manera el taladro.



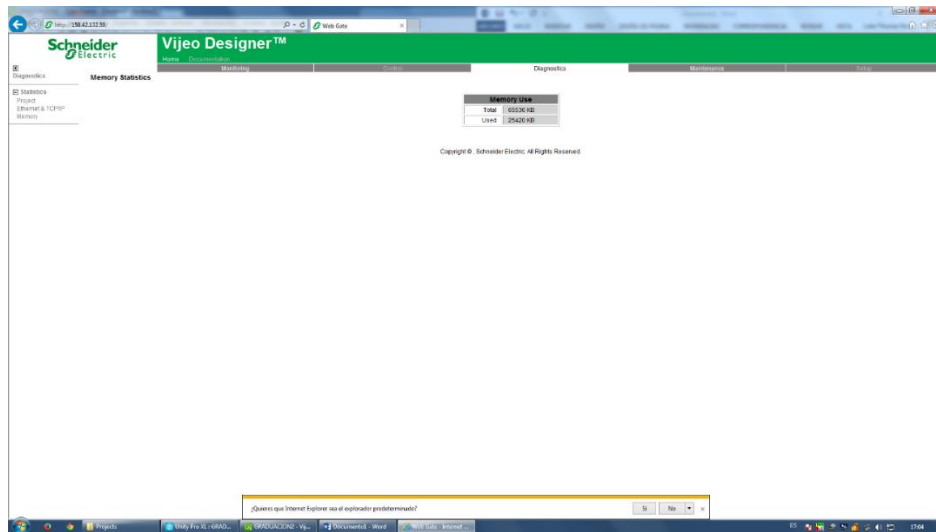
En la siguiente página podemos observar nuestros parámetros de Ethernet, como la IP utilizada, la máscara de red y también la dirección MAC de nuestro dispositivo HMI.



En esta página se puede ver algunos parámetros como el tipo de pantalla táctil que estamos utilizando, el nombre del proyecto, el destino del proyecto, además de las diferentes versiones de software utilizadas.



Finalmente, en esta última página se puede observar la memoria utilizada para el proyecto Vijeo .



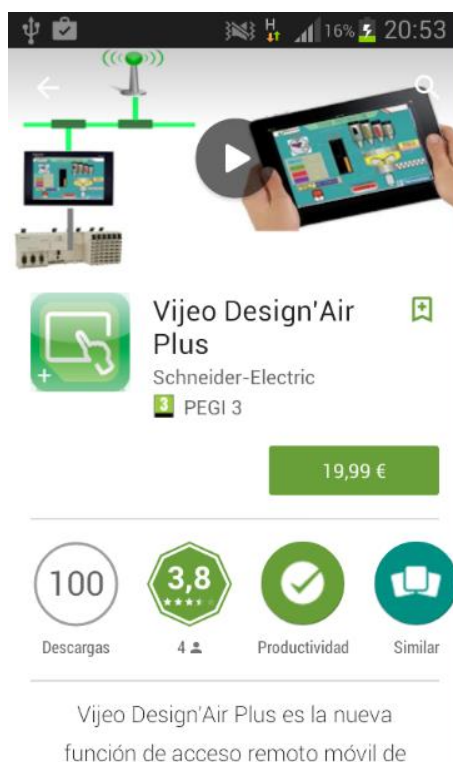
11. Vijeo Design Air Plus

Es una nueva app que ha creado Schneider electric para tener un control remoto a través de una Tablet o smartphones.

Activando el runtime de la aplicación un operador podrá acceder a la representación gráfica de nuestro HMI y mostrar los datos y control de la automatización.

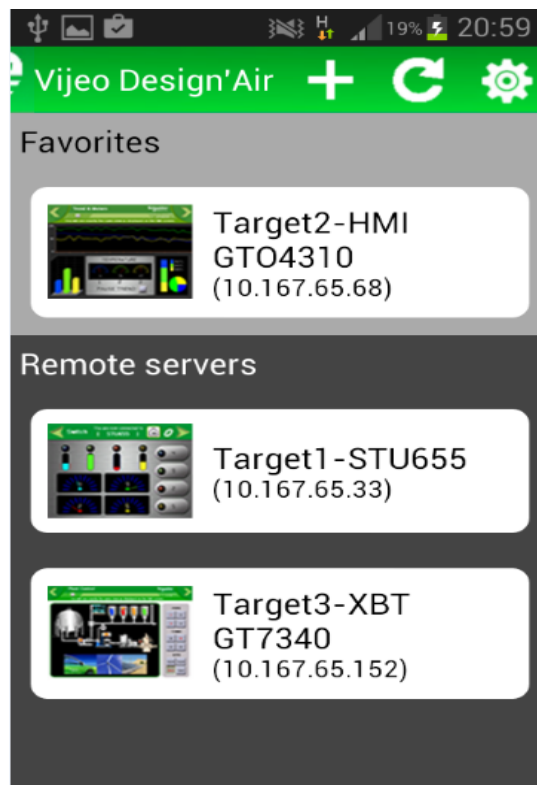
La aplicación Plus además permite la modificación de los paneles gracias a las herramientas gráficas de las que dispone. Aunque esto no se ha mostrado en el proyecto debido a que la app requiere un gasto monetario. Se ha descargado la demo, y indica que con la demo no se nos permite la conexión con el HMI.

El primer paso que debemos realizar es entrar en Play Store de android y descargar la siguiente aplicación.

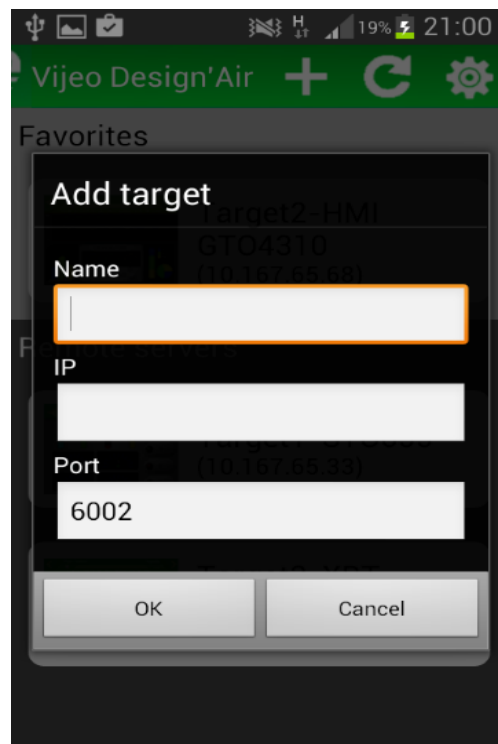


Como se ve en la imagen, la aplicación cuesta 19,99€, hasta hace nada esta aplicación tenía el precio de 99,99€. Por lo visto se ha debido de reducir el precio por las pocas descargas obtenidas.

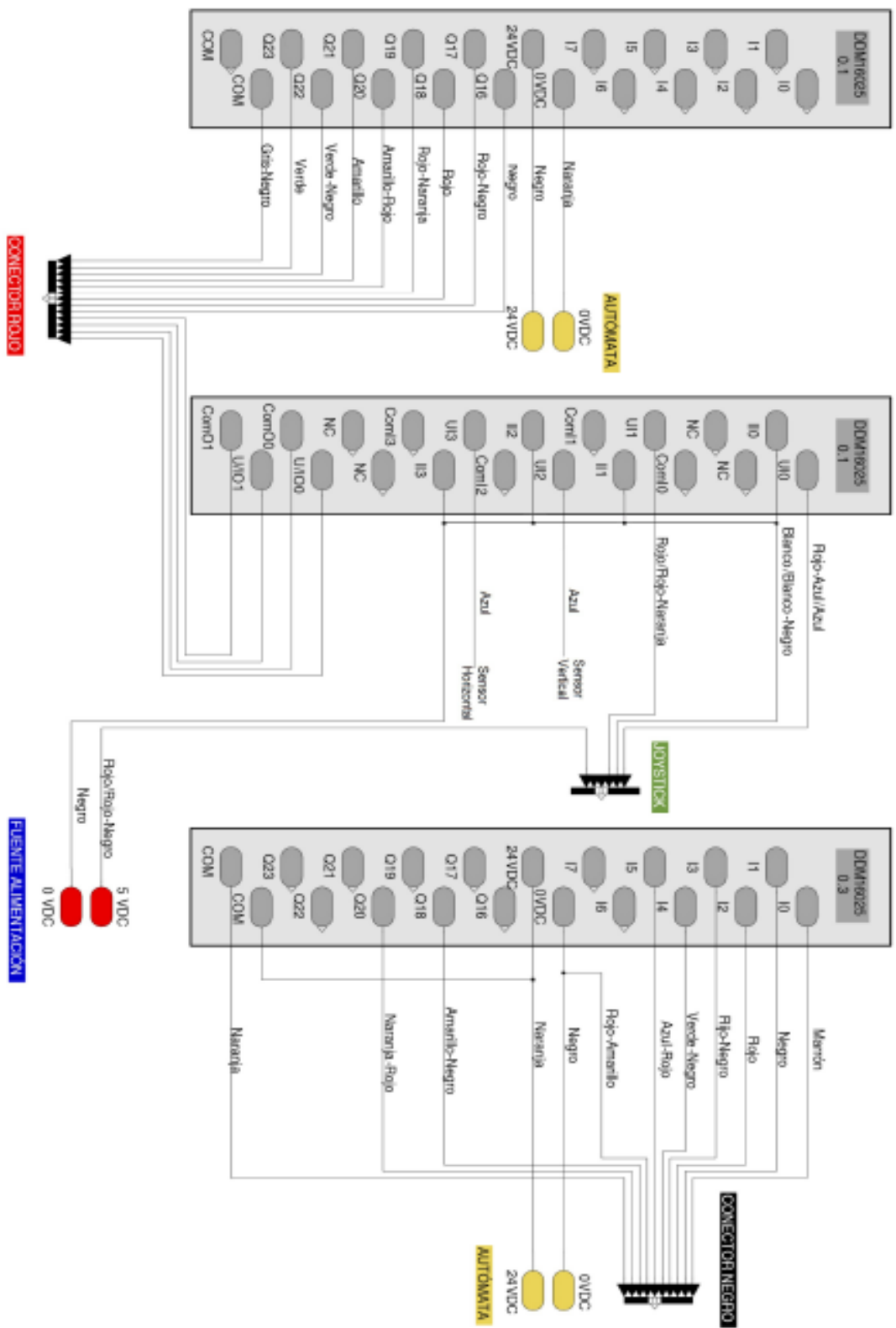
Al abrir la aplicación lo siguiente que se nos muestra es:



Tiene una manejo muy sencillo, para añadir nuestro HMI MAgelis, solo tendremos que darla al signo "+" y nos aparecerá la imagen siguiente:



Como vemos en el panel, solo debemos de insertar el nombre que queremos asignar a nuestro HMI para el dispositivo android, la IP del HMI de Schneider y por último, tendremos que escoger un número de puerto, en el caso de este proyecto se ha utilizado el puerto 6000, que es el puerto predeterminado que asigna el Vijeo Designer.



13.Referencias

Manual de formación Vijeo Designer
Manual de Unity Pro XL v8.0