

8. Autómata programable: MicroLogix 1500

Para realizar el control de las prácticas remotas se hizo primeramente un estudio de mercado para obtener la mejor elección según los objetivos fijados.

Era preciso un autómata programable con al menos 12 salidas de tipo relé y con acceso a Internet. Debía poder ser programable con los lenguajes de programación que siguen el estándar 1131-3 de la IEC (*Internacional Electrotechnical Commission*) para cumplir su función educativa, ya que estos son los utilizados en ambientes industriales. También debía ser económico, de tal manera que los países para los que está destinado el proyecto: Cuba, Curaçao, ... con pocos recursos económicos, pudieran adquirir el material y crear su propio laboratorio remoto.

Actualmente no existen autómatas con esas características a bajo precio, así que se utilizó el material existente en el departamento de Ingeniería Eléctrica de la ETSEIB. Éste era un MicroLogix 1500, una plataforma de control lógico programable de Allen Bradley con el complemento del módulo ENI (Ethernet Network Interface) que lo dota de conectividad a Ethernet (capítulo 9). Está formado por los siguientes componentes esenciales:

- Fuente de alimentación 120/240 V_{CA}
- Circuito de entrada (12 entradas a 24 V_{CC})
- Circuito de salida (12 salidas de relé)
- Procesador modelo 1764-LSP, con 7 Kbit de capacidad para programa de usuario.



Figura 8.1: MicroLogix 1500 de Allen Bradley [10]

Para programar el controlador se usa el software RSLogix 500 de Rockwell Software (capítulo 10.2)



Mediante el puerto RS-232 se acopla el módulo ENI, que le da conectividad a Internet. Este puerto utiliza el protocolo de comunicación DF1 Full-duplex, muy útil cuando se requiere comunicación RS-232 punto a punto. Este protocolo acepta transmisiones simultáneas entre dos dispositivos en ambas direcciones. El protocolo DF1 controla el flujo de mensajes, detecta y señala errores y efectúa reintentos si se detectan errores.

La velocidad de transmisión del puerto por defecto es de 19200 Bps. Este dato deberá ser tomado en cuenta más tarde en la configuración del módulo ENI. Para que no existan problemas de comunicaciones, los dos dispositivos deberán tener la misma velocidad de transmisión.

Funciones de seguridad

Actualmente cualquier estudiante que tenga el software necesario para configurar el autómata y la red de comunicaciones (RSLogix 500 y RSLinx respectivamente), junto con la dirección IP del módulo ENI, puede descargar sus programas.

Uno de los conceptos que ha presentado más problemas en la realización del proyecto es la función de seguridad. Se pretendía realizar un sistema de seguridad que regulara el tráfico de estudiantes con acceso a la planta remota de prácticas. De forma que se asignara un tiempo determinado (10 minutos) a cada estudiante para evaluar el programa realizado, de manera que nadie más pudiera conectarse.

Se estudiaron diferentes alternativas que se fueron descartando por su ineficacia o alto coste.

En primer lugar se pensó en el software de Rockwell: *Security Server*. Esta aplicación bloquea mediante una contraseña el software de programación en el ordenador personal en la que está instalado. Está pensado para restringir el acceso a los programadores de los PLC en un ambiente industrial. En este caso no interesaba ya que es el propio estudiante quien debería programar la aplicación, y una vez tuviera la contraseña podría acceder al autómata en cualquier momento. Por tanto se descartó la opción.

Más tarde se pensó en la opción disponible en el RSLogix de adjudicar una contraseña al propio archivo de programa que se quiere enviar. Este método tampoco es válido ya que se puede anular simplemente borrando la memoria del controlador.

También se planteó la posibilidad de restringir el acceso a la página web que muestra las imágenes de la cámara web. Con esto se impide la visualización del funcionamiento de las prácticas pero no la descarga de los programas al autómata, por tanto se desestimó esta opción.



Como última posibilidad se pensó en la colocación de un PLC que regulase los accesos al PLC de control. Es la única opción que se apuntó como viable pero debido a las actuales dimensiones reducidas del laboratorio remoto no era económicamente procedente. Si el laboratorio creciera en número de plantas entonces sería más conveniente la colocación de un PLC maestro que permitiera el acceso a los diferentes PLC de control de las plantas de prácticas.

Así que finalmente se ha separado como línea futura de trabajo del proyecto actual.



9. Módulo ENI (*Ethernet Network Interface*)

9.1. Función

El módulo 1761-NET-ENI (figura 9.1) proporciona conectividad mediante el protocolo de comunicación Ethernet/IP (*Ethernet Industrial Protocol*) a dispositivos con puerto RS-232 que usen el protocolo de comunicación DF1 full-duplex, en nuestro caso el autómata MicroLogix 1500. Este módulo permite conectar los autómatas programables a una red Ethernet, cargar y descargar programas, intercambiar información entre dispositivos y editar programas lógicos entre otras opciones. [11]



Figura 9.1: Aspecto exterior del módulo ENI [11]

La función del módulo es la de recibir un mensaje del autómata (en formato DF1/PCCC) por el puerto RS-232 y encapsularla poniéndole un envoltorio CIP, este nuevo mensaje se envía a través del puerto Ethernet hacia su destino. El procedimiento es el inverso para un mensaje recibido desde Ethernet, la función será desencapsularlo revelando el mensaje original en formato DF1/PCCC y enviarlo por el puerto RS-232 hacia el autómata (figura 9.2). [11]

Con PCCC (*Programmable Controller Communications Commands*) se entiende los comandos de comunicación de un controlador programable. Para que este formato se pueda enviar a través de la red LAN, Rockwell ha desarrollado el CIP (*Control and Information Protocol*). [11]

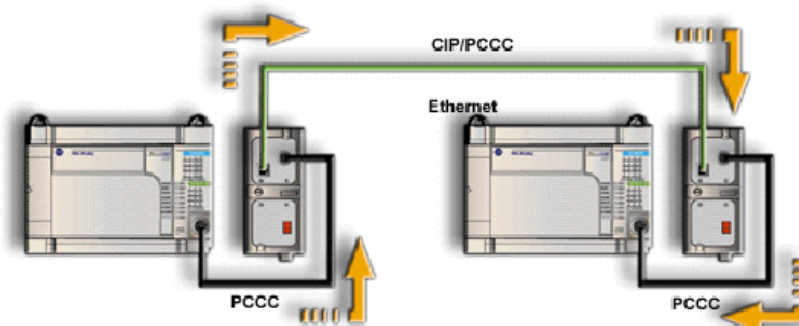


Figura 9.2: Encapsulamiento de los mensajes [11]



9.2. Protocolos de comunicación utilizados

Ethernet es una LAN que transmite información a velocidades de 10 o 100 millones de bits por segundo (Mbps). En 1985 se estandarizó con el nombre de: "IEEE 802.3 *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*" o más escuetamente "IEEE 802.3 CSMA/CD", lo que significa: [13]

Carrier Sense: cada nodo escucha continuamente el tráfico existente en el medio para determinar cuando aparece un hueco en el que no exista información. [12]

Multiple Access: el nodo empieza a transmitir cuando se detecta que no existe tráfico. [12]

Collision Detection: Si dos o más nodos en un dominio de colisión empiezan a transmitir al mismo tiempo, las cadenas de bits colisionarán y las dos conexiones se invalidarán. Los nodos deben ser capaces de detectar la colisión antes de acabar de enviar su información y detener la transmisión. Entonces esperan un tiempo aleatorio antes de volver a probar la transmisión. Y así sucesivamente. [12]

Este funcionamiento hace que el protocolo no sea determinista, es decir, que no se pueda asegurar el envío de la información aunque la posibilidad de que no se envíe es ínfima. De todas maneras existen vías para garantizar ese determinismo y poder utilizar el protocolo en ambientes industriales. Una de ellas sería el uso de un *switch* de Ethernet para cada nodo más la utilización del protocolo *full-duplex*. El nodo únicamente se comunica con el *switch* y existen dos sentidos de comunicación debido al *full-duplex*, por lo que no hay manera de colisionar, aunque es una medida con un precio elevado. Un *switch* es un dispositivo de interconexión entre estaciones de una misma red.

En una red Ethernet cada nodo conectado al sistema funciona independientemente de los demás, es decir, no existe un controlador central y todos los elementos del sistema se conectan a un elemento compartido (el medio físico), por donde se transporta la señal. En este caso el medio físico consiste en un cable de par trenzado. [13]

Ethernet ocuparía las dos primeras capas del modelo de referencia ISO-OSI (figura 9.3). Éste modelo se creó a partir de la necesidad de crear un estándar para las comunicaciones asegurando la conectividad a todos los niveles (físico, datos, aplicaciones...). (Anexo A.4)



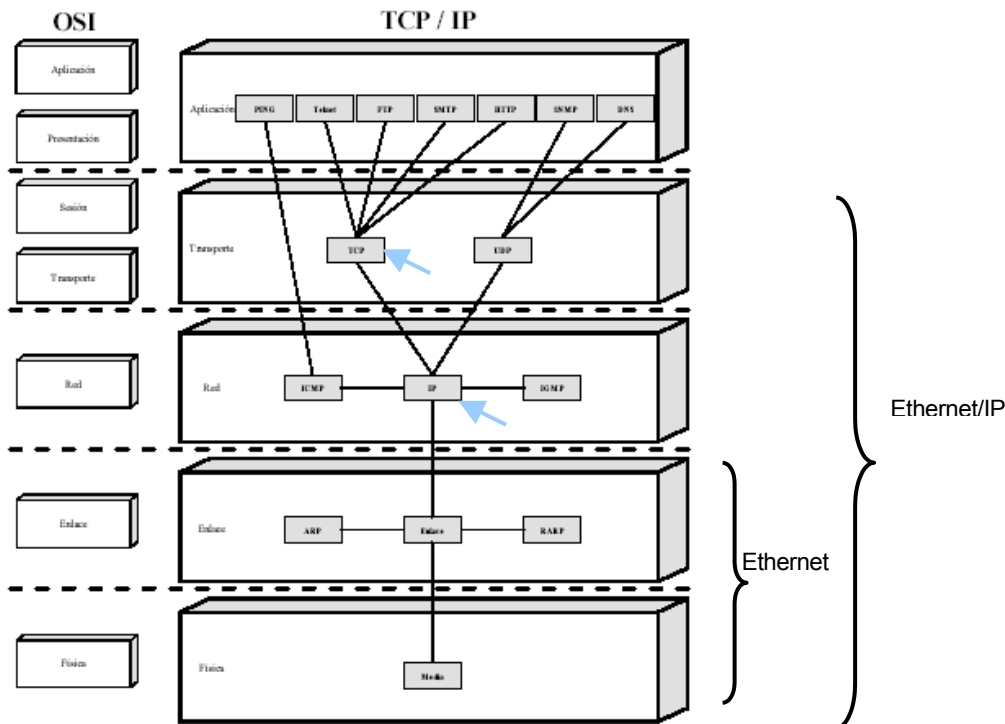


Figura 9.3: Correspondencia capas OSI [14]

Para comunicar un nodo de una LAN con un nodo de otra LAN se tiene que usar un protocolo de comunicaciones entre LANs, el *Internet Protocol* (IP). IP es un protocolo *connectionless* y no asegura la entrega correcta del mensaje en el destino, esa seguridad deben proporcionarla protocolos de capas superiores. El protocolo más utilizado en este caso es el TCP (*Transport Control Protocol*). Este conjunto de protocolos forman el llamado Ethernet/IP (*Ethernet Industrial Protocol*) (figura 9.3). [13]

9.3. Configuración del módulo

Para configurar el módulo se ha usado el software ENI *Configuration Utility* de Rockwell Software, descargable de manera gratuita en la página web de Allen Bradley. Existe otra opción de configuración por medio de un mensaje a un nodo preconfigurado (255) pero esta es más complicada y por tanto se ha optado por el ENI *Configuration Utility*.

Para realizar la configuración se debe conectar el módulo al puerto serie del ordenador personal mediante el puerto RS-232. Se precisa de alimentación externa al módulo, ya que normalmente éste toma la alimentación de la fuente del autómata mediante el puerto RS-232, que está ocupado por la comunicación con el PC.



Para tal efecto se usará una fuente de alimentación Hyperion (figura 9.4) que proporciona el voltaje de 24 V necesario para el correcto funcionamiento del módulo.

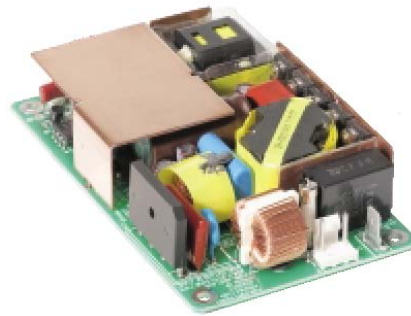


Figura 9.4: Fuente de alimentación [15]

Se debe revisar en primera instancia que el puerto serie del ordenador esté libre y tenga las características adecuadas para comunicarse con el módulo.

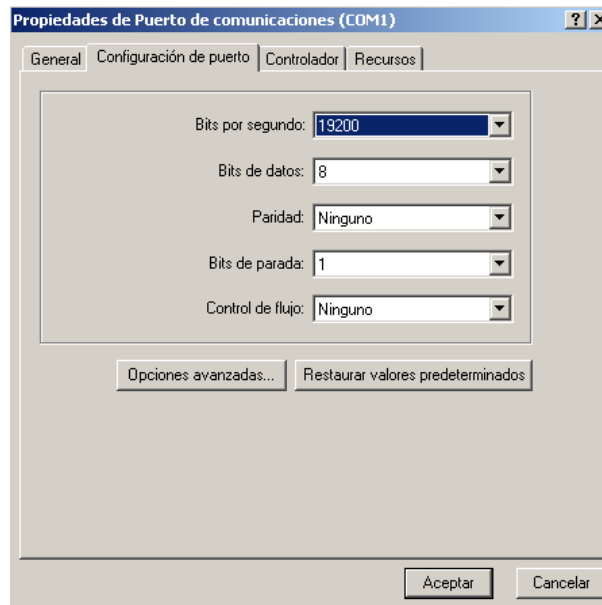


Figura 9.5: Configuración del puerto serie del ordenador

La velocidad de transferencia ha de ser la misma que se ha configurado en el MicroLogix 1500 y la que se configurará en el módulo. Se ha escogido 19200 Bps, velocidad más que suficiente para la transmisión de los datos del programa.

Una vez asegurada la buena comunicación y la alimentación del módulo se procede a la propia configuración del mismo. Primeramente se definen los parámetros del puerto serie del ordenador que se va a utilizar:

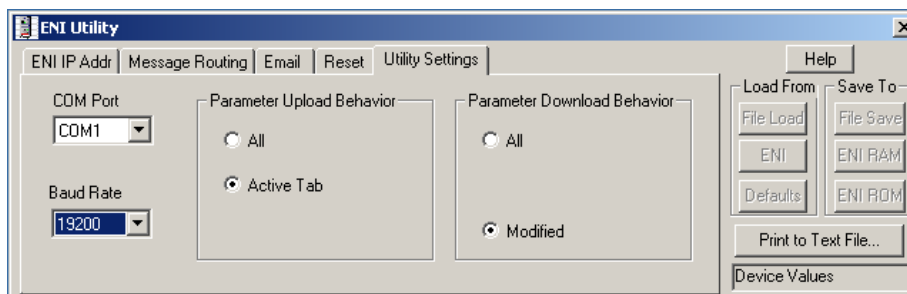


Figura 9.6: Definición de los parámetros de conexión con el ordenador



A continuación se configura la dirección IP del módulo ENI. Debemos introducir los valores de la dirección IP que se le asignará al módulo junto con los parámetros de la red en la que se encontrará (máscara de subred y *Gateway*).

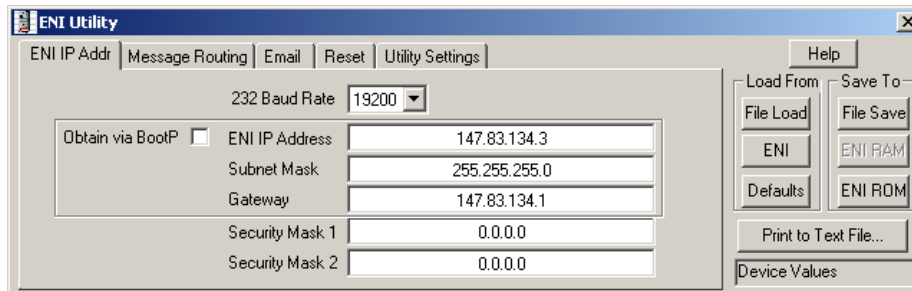


Figura 9.7: Definición de la dirección IP del módulo

La máscara de subred se usa para interpretar las direcciones IP cuando la red está dividida en subredes, como es nuestro caso. Y Gateway o pasarela es la dirección del dispositivo que realiza la unión entre la red LAN y Internet. En la figura 7.2 se muestra la configuración de comunicación existente en el laboratorio remoto.

Finalmente, una vez configurados los parámetros anteriores, se carga a la memoria ROM del módulo para que se quede de forma permanente.

Existen otras posibilidades de configuración en este software como son: configuración de las IPs de otros nodos de la red Ethernet y configuración de direcciones de correo electrónico para informar de alarmas o estados. En el caso del presente curso estas opciones no son necesarias, pero se deben tener en cuenta para futuras ampliaciones.

Una vez finalizados los pasos anteriores el módulo está listo para su conexión con el MicroLogix 1500 a través del puerto RS-232. Solo restará la configuración de la red y el PLC mediante el software específico (capítulo10).



10. Software de programación

10.1. RSLinx

RSLinx es un sistema operativo de red (*Network Operating System*) que se encarga de regular las comunicaciones entre los diferentes dispositivos de la red. Proporciona el acceso de los controladores Allen-Bradley a una gran variedad de aplicaciones de *Rockwell Software*, tales como RSLogix 500.

Existen diferentes paquetes según la funcionalidad a la que se haya de destinar el software. En nuestro caso y debido a las necesidades del proyecto se usará RSLinx Lite, que es el paquete más sencillo y económico. Éste no está disponible comercialmente pero se incluye en el RSLogix 500, el software propio de programación del autómatas. A partir de ahora nos referiremos al producto únicamente como RSLinx.

Primeramente se debe configurar la red de comunicaciones de nuestros dispositivos. Para ello se debe configurar el controlador óptimo, que en este caso se trata de una red con dispositivos conectados a Ethernet (*Ethernet devices*).

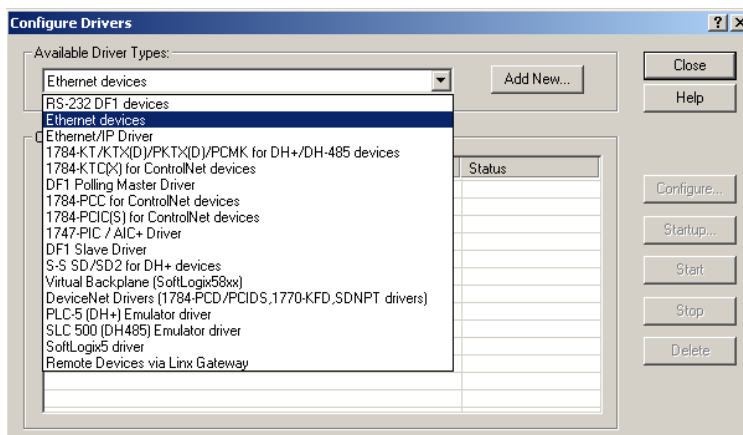


Figura 10.1: Elección del controlador

Un controlador es la interface de software al dispositivo de hardware, que en este caso es el módulo ENI, y permite la conexión con el RSLinx.

Se especificará a continuación la dirección IP del módulo ENI (figura 10.2) que es el único elemento de nuestra red, ya que el ordenador personal del usuario viene representado por el propio RSLinx del usuario. La dirección IP será la que se ha asignado previamente con el *ENI Configuration Utility*.



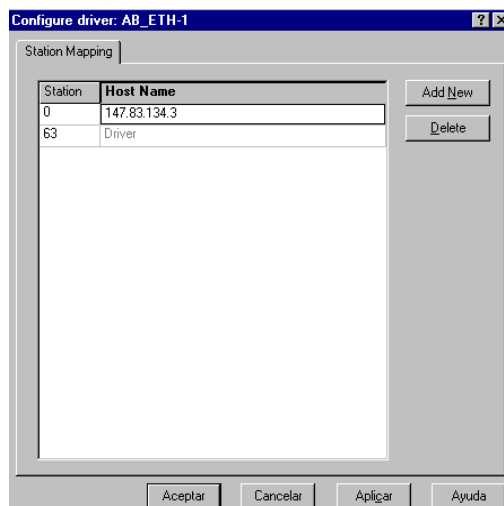


Figura 10.2: Definición de los componentes de la red

Una vez está bien configurada se obtiene la siguiente imagen, de la red funcionando (figura 10.3). Como se puede ver a partir de la dirección IP del módulo ENI el software ha detectado el controlador MicroLogix 1500 que está conectado a él. La red está configurada.

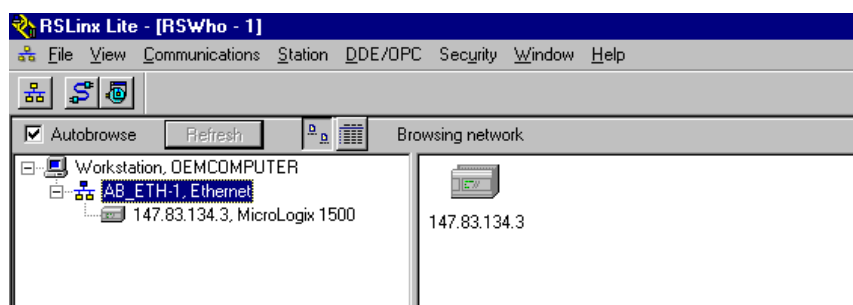


Figura 10.3: Pantalla principal de RSLinx Lite

10.2. RSLogix 500

RSLogix 500 es el software destinado a la creación de los programas del autómata en lenguaje de esquema de contactos o también llamado lógica de escalera (*Ladder*).

Existen diferentes partes dentro de la aplicación, las más importantes son: el editor *Ladder*, el panel de resultados (donde se genera la lista de errores por medio del verificador de proyectos) y el árbol de proyectos mediante el cual se manejan todas las posibilidades existentes.

Este producto se ha desarrollado para funcionar bajo el sistema operativo Windows®.



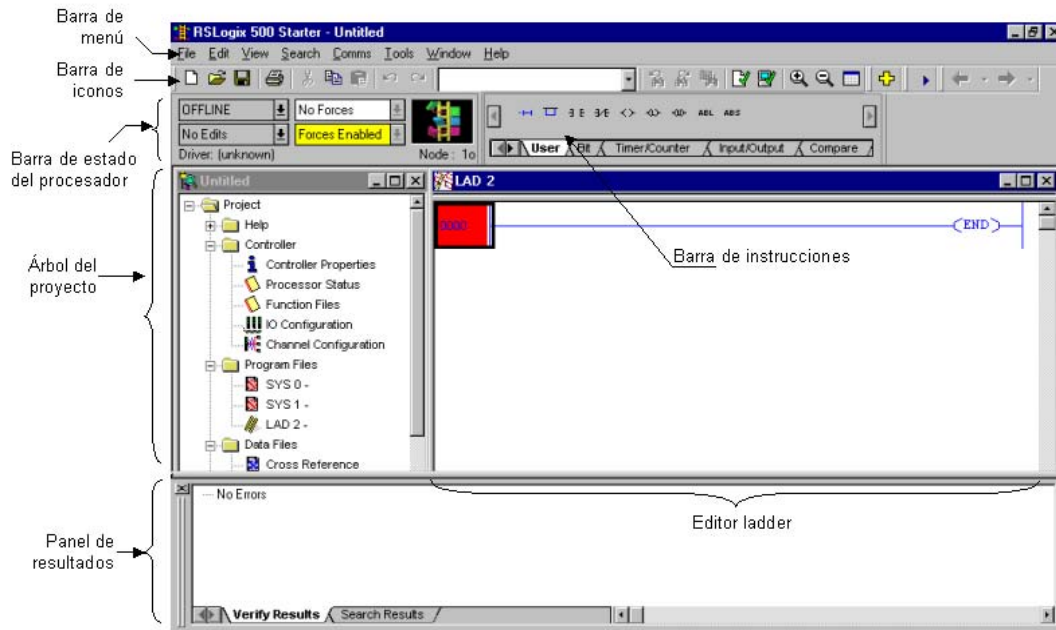


Figura 10.4: Pantalla principal del RSLogix 500

Al iniciar un nuevo proyecto con el RSLogix 500 se debe especificar el tipo de autómeta utilizado y la red de comunicación a la que pertenece. Escogemos el MicroLogix 1500 LSP series C y lo cargamos.

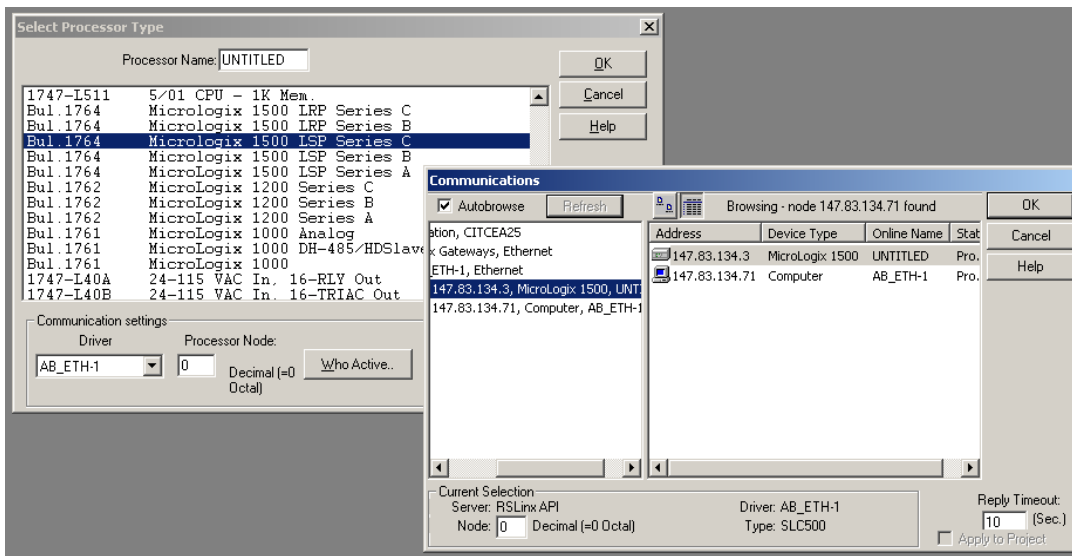


Figura 10.5: Selección del procesador

Si la red se ha configurado de manera correcta anteriormente en el RSLinx, ésta ya aparecerá como opción de *driver* en el menú desplegable de *Communications settings* (figura 10.5). Si no apareciera se puede acceder también a las redes existentes mediante el



pulsador *Who Active* o desde el apartado *Controller Properties* del árbol de proyecto (figura 10.4), apareciendo una ventana que se muestra en la figura 10.6.

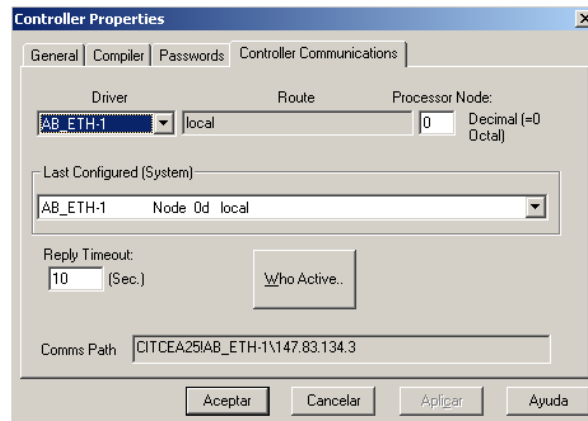


Figura 10.6: Configuración de las comunicaciones del autómata

Si no apareciese la red especificada debe volver a configurarse mediante RSLinx.

Una vez configurados estos parámetros, el alumno será capaz de enviar el programa al autómata, interactuar con él mediante el forzado de variables y visualizar los resultados obtenidos mediante las imágenes de la cámara web (figura 10.7). Para ello solo debe desplegar las opciones de la barra de estado del procesador (figura 10.4) y pasar del modo *offline* a *download*. Con esto el programa ya se enviará al autómata. Se deben aceptar las distintas peticiones que realiza el software hasta llegar al modo *remote run*, es decir, funcionamiento remoto. Este procedimiento está ampliamente explicado en el Anexo B.4, con la explicación de los problemas usuales que se pueden encontrar.

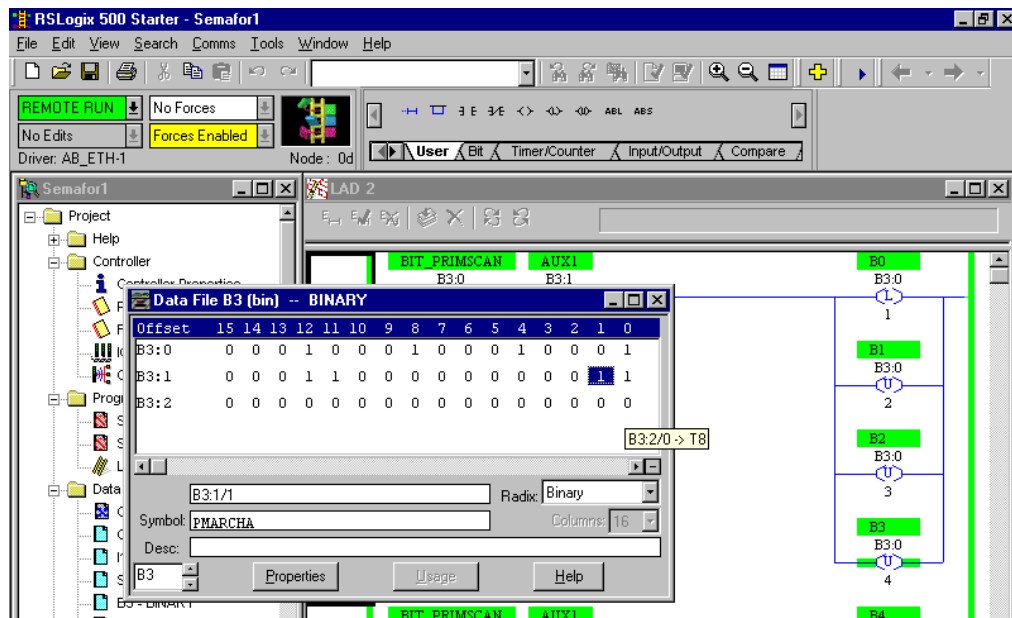


Figura 10.7: Programa *on-line* y forzado de entrada

